

PEMBANGUNAN SISTEM INFORMASI PASIEN OPERASI DAN VISITE UNTUK TENAGA MEDIS BERBASIS ANDROID PADA RUMAH SAKIT XYZ

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Eki Yusandhi Iskandar
NIM: 145150401111036



**PROGRAM STUDI SISTEM INFORMASI
SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI PASIEN OPERASI DAN VISITE UNTUK
TENAGA MEDIS BERBASIS ANDROID PADA RUMAH SAKIT XYZ

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Eki Yusandhi Iskandar
NIM: 145150401111036

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Herman Tolle S.T, M.T
NIP: 19740823 200012 1 001

Issa Arwani S.Kom, M.Sc
NIP: 19830922 201212 1 003

Mengetahui
Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle S.T, M.T
NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Agustus 2018

Eki Yusandhi Iskandar

NIM: 145150401111036



Curriculum Vitae

Data Pribadi

Nama : Eki Yusandhi Iskandar
Tempat, Tanggal lahir : Padang Sidempuan , 18 Maret 1996;
Agama : Islam;
Alamat rumah : Jl. Soekarno Hatta, Kompleks Griya
Shanta Eksekutif Blok P
No 421 , Malang
Nomer telepon : 081212488321 (*mobile phone*);
Email : eki_iskandar@yahoo.com

Riwayat Pendidikan

- **Pendidikan Formal:**
- 2012 sampai dengan 2014 : SMA Negeri 1 Kota Serang
- 2011 sampai dengan 2011 : SMA Negeri 1 Medan (Kelas Internasional)
- 2008 sampai dengan 2011 : SMP Panca Budi Medan
- 2005 sampai dengan 2008 : SD Panca Budi Medan
- 2004 sampai dengan 2004 : SD Negeri 1 Banda Aceh
- 2003 sampai dengan 2004 : SD Pertiwi Medan
- 2002 sampai dengan 2003 : SD Pertiwi Padang

Riwayat Organisasi

- 2014 sampai dengan sekarang : Anggota Persatuan Mahasiswa Banten Brawijaya
- 2008 sampai dengan 2011 : Anggota OSIS SMP Panca Budi Medan

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT karena atas berkah dan rahmat-Nya skripsi yang berjudul “Pembangunan Sistem Informasi Pasien Operasi dan Visite untuk Tenaga Medis Berbasis Android pada Rumah Sakit XYZ” dapat terselesaikan

Banyak pihak yang telah memberi bantuan baik moral maupun material baik secara langsung maupun tidak langsung untuk terselesaikannya skripsi ini, oleh karena itu terima kasih sebesar-besarnya penulis ucapkan kepada:

1. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
2. Dr. Eng. Herman Tolle S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya dan selaku dosen pembimbing satu yang telah memberikan bantuan, arahan, dan bimbingan untuk terselesaikannya skripsi ini.
3. Suprpto S.T., M.T selaku Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
4. Issa Arwani S.Kom, M.Sc selaku pembimbing dua yang telah memberikan bantuan, arahan, dan bimbingan untuk terselesaikannya skripsi ini.
5. Anggota satuan kerja IT Rumah Sakit XYZ yang membantu dalam proses identifikasi permasalahan dan analisis kebutuhan untuk skripsi ini.
6. Responden tenaga medis yang membantu proses pengujian skripsi ini.
7. Orang tua, keluarga, dan sanak saudara yang memberikan dukungan do’a dan moral untuk terselesaikannya skripsi ini.
8. Seluruh pihak yang berperan dalam membantu proses terselesaikannya skripsi ini yang tidak dapat disebutkan seluruhnya satu per satu.

Penulis menyadari bahwa skripsi ini tidaklah sempurna dan menyadari sepenuhnya bahwa ketidaksempurnaan skripsi ini adalah karena keterbatasan materi dan ilmu yang dimiliki penulis. Dengan segala keterbatasan skripsi ini, penulis berharap semoga ada manfaat dan kegunaan bagi pembaca dan bagi kemajuan ilmu pengetahuan khususnya bidang ilmu komputer.

Malang, 1 Agustus 2018

Penulis

eki_iskandar@yahoo.com

ABSTRAK

Dalam institusi kesehatan publik, alur informasi merupakan salah satu hal terpenting, terutama bagi tenaga medis. Informasi pasien, ruang rawat inap pasien, dan jadwal operasi merupakan beberapa dari sekian banyak informasi penting yang harus didapatkan oleh tenaga medis rumah sakit. Tidak terkecuali Rumah Sakit XYZ, dengan jumlah tenaga medis lebih dari 200 orang, alur informasi dari dan kepada tenaga medis haruslah dapat diakses dengan cepat dan akurat. Akan tetapi belum ada sebuah media bagi tenaga medis Rumah Sakit XYZ untuk dapat dengan mudah mengakses informasi-informasi dasar tentang pasien rawat inap, proses visite, dan operasi.

Oleh karena itu, untuk membantu meningkatkan proses bisnis yang ada pada Rumah Sakit XYZ, dan untuk memberikan gambaran tentang pengimplementasian solusi untuk institusi kesehatan lain dengan proses bisnis yang sama, dibangunlah sistem informasi pasien, visite dan operasi bagi tenaga medis berbasis Android. Hasil implementasi aplikasi ini mencakup terpenuhinya seluruh kebutuhan fungsional sistem berdasarkan hasil analisis kebutuhan dan tingkat usability dengan skor 79.5 berdasarkan hasil survey responden dengan metode SUS(System Usability Scale), yang berarti sudah berada dalam kriteria "Dapat Diterima Pengguna".

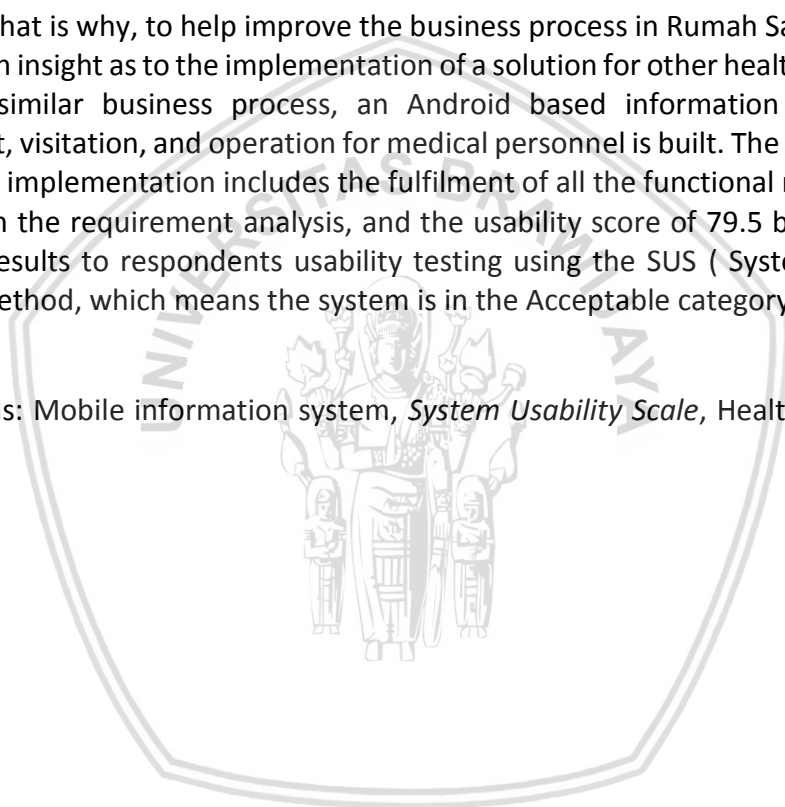
Kata kunci: Sistem informasi *mobile*, *System Usability Scale*, Aplikasi institusi kesehatan

ABSTRACT

In a public health institution, information flow is one of the most important things, especially to the medical personnel. Patient information, inpatient room information, and operation schedule are some of the most crucial information that a medical personnel has to acquire. Rumah Sakit XYZ is no exception, with a total of more than 200 medical personnel, the information flow to and from the medical person must be able to be accessed quickly and accurately. However there is not yet a platform for the medical personnel of Rumah Sakit XYZ to easily access basic information such as inpatient information, visitation, and operation.

That is why, to help improve the business process in Rumah Sakit XYZ, and to give an insight as to the implementation of a solution for other health institution with a similar business process, an Android based information system for inpatient, visitation, and operation for medical personnel is built. The result of this system's implementation includes the fulfilment of all the functional requirement based on the requirement analysis, and the usability score of 79.5 based on the survey results to respondents usability testing using the SUS (System Usability Scale) method, which means the system is in the Acceptable category.

Keywords: Mobile information system, *System Usability Scale*, Health institution system



DAFTAR ISI

PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	v
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR tabel	xii
DAFTAR GAMBAR.....	xiii
DAFTAR lampiran	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Model <i>Waterfall</i>	5
2.3 Diagram Kelas (<i>Class Diagram</i>)	6
2.4 Sistem Operasi Android	6
2.4.1 Komponen Aplikasi Android.....	6
2.4.2 Daur Hidup <i>Activity</i> pada Aplikasi Android	7
2.4.3 Daur Hidup <i>Service</i> pada Aplikasi Android.....	8
2.5 <i>System Usability Scale (SUS)</i>	8
BAB 3 METODOLOGI	10
3.1 Tipe dan Metode Penelitian	10
3.2 Lokasi Penelitian	10
3.3 Metode Pengumpulan data	10
3.4 Sistematika Penelitian	10

3.4.1 Identifikasi Permasalahan	10
3.4.2 Pengkajian Pustaka	10
3.4.3 Analisis Proses Bisnis.....	11
3.4.4 Analisis Kebutuhan.....	11
3.4.5 Perancangan.....	12
3.4.6 Implementasi	12
3.4.7 Pengujian dan Analisis Hasil.....	12
3.4.8 Kesimpulan Hasil dan Saran	12
BAB 4 Analisis dan perancangan	13
4.1 Analisis Kebutuhan	13
4.1.1 Analisis Kebutuhan Sistem	13
4.1.2 Gambaran Sistem	17
4.1.3 Analisis Kebutuhan Fungsional	17
4.1.4 Analisis Kebutuhan Non-Fungsional	19
4.1.5 Use Case	19
4.1.6 Use Case Scenario	19
4.2 Perancangan	24
4.2.1 Sequence Diagram	24
4.2.2 Class Diagram	30
4.2.3 Perancangan Antar Muka.....	31
4.2.4 Perancangan Screenflow	38
4.2.5 Perancangan Data	39
BAB 5 Implementasi	40
5.1 Implementasi	40
5.2 Spesifikasi Perangkat Lunak.....	40
5.3 Spesifikasi Perangkat Keras	40
5.4 Batasan Implementasi	40
5.5 Implementasi Basis Data.....	41
5.5.1 Dokter.....	41
5.5.2 Pasien	41
5.5.3 Ruang.....	42
5.5.4 Rawat Inap	42

5.5.5 Operasi	43
5.6 Implementasi Class	43
5.7 Implementasi Antar Muka	44
5.7.1 Halaman Login.....	45
5.7.2 Halaman Menu.....	45
5.7.3 Halaman Daftar Pasien.....	46
5.7.4 Halaman Daftar Visite	46
5.7.5 Halaman Daftar Operasi.....	47
5.7.6 Halaman Detil Pasien	47
5.7.7 Halaman Detil Visite	48
5.7.8 Halaman Detil Operasi	48
5.7.9 Halaman Detil Operasi 2	49
5.8 Implementasi Kode	49
5.8.1 Fungsi Login.....	49
5.8.2 Fungsi Daftar Pasien.....	50
5.8.3 Fungsi Daftar Visite	50
5.8.4 Fungsi Daftar Operasi.....	50
5.8.5 Fungsi Detil Pasien	51
5.8.6 Fungsi Detil Visite.....	51
5.8.7 Fungsi Detil Visite	52
5.8.8 Fungsi Konfirmasi Visite	52
5.8.9 Fungsi Status Operasi	52
BAB 6 Pengujian & analisis.....	54
6.1 Pengujian	54
6.2 Pengujian Fungsional	54
6.2.1 Kasus Uji Login.....	54
6.2.2 Kasus Uji Melihat Daftar Pasien	55
6.2.3 Kasus Uji Melihat Daftar Visite.....	55
6.2.4 Kasus Uji Melihat Daftar Operasi	56
6.2.5 Kasus Uji Melihat Detil Pasien.....	56
6.2.6 Kasus Uji Melihat Detil Visite	57
6.2.7 Kasus Uji Melihat Detil Operasi.....	57

6.2.8 Kasus Uji Konfirmasi Visite	58
6.2.9 Kasus Uji Status Operasi.....	58
6.3 Hasil Pengujian Fungsional	59
6.4 Pengujian Usabilitas.....	59
6.5 Hasil Pengujian Usabilitas	61
6.6 Analisis Hasil Pengujian Usabilitas	62
6.7 Pengujian Waktu.....	62
BAB 7 Kesimpulan	64
7.1 Kesimpulan.....	64
7.2 Saran	64
DAFTAR PUSTAKA.....	65



DAFTAR TABEL

Tabel 4.1 Kebutuhan fungsional	15
Tabel 4.2 Skenario masuk	17
Tabel 4.3 Skenario melihat daftar pasien	18
Tabel 4.4 Skenario melihat detil pasien	19
Tabel 4.5 Skenario melihat jadwal visite.....	19
Tabel 4.6 Skenario melihat detil visite	20
Tabel 4.7 Skenario melihat jadwal operasi	20
Tabel 4.8 Skenario melihat detil operasi.....	21
Tabel 4.9 Skenario mengonfirmasi visite	21
Tabel 4.10 Skenario mengubah status operasi	21
Tabel 5.1 Spesifikasi perangkat lunak	38
Tabel 5.2 Spesifikasi perangkat keras	38
Tabel 5.3 Tabel dokter	39
Tabel 5.4 Tabel pasien.....	39
Tabel 5.5 Tabel ruang.....	40
Tabel 5.6 Tabel rawat inap.....	40
Tabel 5.7 Tabel operasi	41
Tabel 6.1 Kasus uji login	52
Tabel 6.2 Kasus uji melihat daftar pasien	52
Tabel 6.3 Kasus uji melihat daftar visite	53
Tabel 6.4 Kasus uji melihat daftar operasi.....	53
Tabel 6.5 Kasus uji melihat detil pasien	54
Tabel 6.6 Kasus uji melihat detil visite	55
Tabel 6.7 Kasus uji melihat detil operasi.....	55
Tabel 6.8 Kasus uji konfirmasi visite	56
Tabel 6.9 Kasus uji status operasi	56
Tabel 6.10 Pertanyaan pengujian <i>usability</i> SUS.....	57
Tabel 6.11 Hasil Pembobotan Jawaban Tiap Responden	59
Tabel 6.12 Hasil Perhitungan Skor SUS	59

DAFTAR GAMBAR

Gambar 2.1 Daur hidup <i>activity</i> pada aplikasi Android	7
Gambar 3.3 Diagram Sistematika Penelitian	11
Gambar 4.1 Alur proses bisnis informasi pasien.....	13
Gambar 4.2 Alur proses bisnis dengan sistem	14
Gambar 4.3 Alur proses bisnis konfirmasi visite	15
Gambar 4.4 Alur proses bisnis konfirmasi visite dengan sistem	15
Gambar 4.5 Alur proses bisnis status operasi.....	16
Gambar 4.6 Alur proses bisnis status operasi dengan sistem	16
Gambar 4.7 Gambaran Sistem	17
Gambar 4.8 Diagram use case	17
Gambar 4.9 Sequence diagram detil pasien	24
Gambar 4.10 Sequence diagram detil visite	25
Gambar 4.11 Sequence diagram daftar operasi.....	26
Gambar 4.12 Sequence diagram detil operasi.....	27
Gambar 4.13 Sequence diagram konfirmasi visite	27
Gambar 4.14 Sequence diagram konfirmasi operasi.....	28
Gambar 4.15 Class diagram aplikasi	29
Gambar 4.16 Rancangan antar muka login.....	30
Gambar 4.17 Rancangan antar muka halaman utama	31
Gambar 4.18 Rancangan antar muka halaman daftar pasien	32
Gambar 4.19 Rancangan antar muka halaman detil pasien.....	32
Gambar 4.20 Rancangan antar muka halaman daftar visite	33
Gambar 4.21 Rancangan antar muka halaman detil visite.....	34
Gambar 4.22 Rancangan antar muka halaman daftar operasi.....	35
Gambar 4.23 Rancangan antar muka halaman detil operasi	36
Gambar 4.24 Screenflow aplikasi.....	36
Gambar 4.25 Struktur data	37
Gambar 5.1 Diagram implementasi kelas	42
Gambar 5.2 Antar muka halaman Login	43
Gambar 5.3 Antar muka halaman Menu	43

Gambar 5.4 Antar muka halaman daftar pasien.....	44
Gambar 5.5 Antar muka halaman daftar visite.....	44
Gambar 5.6 Antar muka halaman daftar operasi	45
Gambar 5.7 Antar muka halaman detil pasien	45
Gambar 5.8 Antar muka halaman detil visite	46
Gambar 5.9 Antar muka halaman detil operasi.....	46
Gambar 5.10 Antar muka halaman detil operasi 2.....	47
Gambar 6.1 Penilaian Skor SUS.....	60



DAFTAR LAMPIRAN

LAMPIRAN A PERTANYAAN DAN HASIL WAWANCARA	63
LAMPIRAN B KUESIONER SUS RESPONDEN	65



BAB 1 PENDAHULUAN

1.1 Latar belakang

Ketepatan dan kecepatan alur informasi dalam layanan kesehatan adalah salah satu hal terpenting yang dapat mempengaruhi kesehatan dan keselamatan seseorang. Salah satu aktor utama yang membutuhkan ketepatan dan kecepatan informasi adalah tenaga medis seperti dokter dan perawat. Tanpa adanya alur informasi yang baik, tenaga medis berisiko untuk mengkompromikan kesehatan pasiennya.

Salah satu layanan kesehatan yang membutuhkan alur informasi yang lebih baik adalah Rumah Sakit XYZ. Jumlah tenaga medis dokter yang tersedia di Rumah Sakit XYZ mencapai lebih dari 200 orang. Setiap tenaga medis dapat bertanggung jawab banyak pasien, oleh karena itu Rumah Sakit XYZ memiliki sistem yang digunakan untuk pencatatan data tenaga medis, seperti data pasien rawat inap, data operasi, dan data visite.

Pengimplementasian sebuah sistem pencatatan data tenaga medis ternyata tidak mempercepat alur informasi di Rumah Sakit XYZ, karena data tersebut hanya bisa diakses secara lokal oleh staff administrasi atau rekam medik yang ada di rumah sakit. Setiap tenaga medis masih harus melakukan pengecekan manual melalui *staff* rumah sakit, untuk mengetahui daftar dan jadwal pasien yang perlu dioperasi atau visite. Selain meningkatkan ketidakefisienan waktu, akses data pasien yang sulit dapat menyebabkan berkurangnya kualitas tenaga medis di mata pasien, atau bahkan berakibat fatal pada kesehatan pasien, jika operasi atau visite yang dilaksanakan bersifat darurat. Keterlambatan waktu operasi atau waktu visite disebabkan oleh perubahan jadwal atau ruangan yang mendadak, proses yang lama untuk mendapatkan informasi ruangan operasi atau visite, dan dapat juga disebabkan oleh lokasi praktik tenaga medis yang berada di beberapa tempat. Waktu tercepat yang dibutuhkan dokter untuk mendapatkan informasi pasien hanya dapat dilakukan melalui aplikasi komunikasi (telepon atau aplikasi pesan daring) dalam waktu 3 hingga 5 menit. Jika media komunikasi tidak dapat dilakukan, dokter harus mengecek secara mandiri data pasien melalui staff rumah sakit, yang menghabiskan waktu 5-15 menit tergantung dari ketersediaan staff dan jumlah pasien yang datanya ingin diakses. Oleh karena itu diperlukan cara untuk mempercepat alur informasi dari bagian rekam medis ke tenaga medis.

Kebutuhan untuk akses data pasien operasi dan rawat inap yang tersimpan di basis data Rumah Sakit XYZ oleh tenaga medis dokter secara *mobile*, untuk mempercepat proses pengecekan data pasien rawat inap merupakan hal yang melandasi dilakukannya penelitian ini. Dengan adanya penelitian ini, diharapkan tenaga medis di rumah sakit dengan proses bisnis yang memiliki kesamaan dengan Rumah Sakit XYZ dapat mempermudah akses data pasien yang akan di operasi dan visite, serta meminimalisir *human error* yang dapat terjadi karena kurangnya informasi yang tersedia untuk tenaga medis.

1.2 Rumusan masalah

Berdasarkan latar belakang, rumusan masalah pada penelitian ini adalah:

1. Bagaimana analisis kebutuhan sistem informasi untuk mempercepat proses pengambilan data pasien operasi dan visite oleh tenaga medis Rumah Sakit XYZ?
2. Bagaimana rancangan sistem informasi pasien operasi dan visite untuk dokter Rumah Sakit XYZ pada perangkat berbasis Android?
3. Bagaimana implementasi sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ pada perangkat berbasis Android?
4. Bagaimana tingkat usabilitas pada sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ?

1.3 Tujuan

Tujuan yang ingin dicapai adalah:

1. Mendapatkan hasil analisis kebutuhan untuk untuk mempercepat proses pengambilan data pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ.
2. Merancang sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ pada perangkat bergerak berbasis Android.
3. Mengimplementasi sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ pada perangkat bergerak berbasis Android.
4. Menguji validitas sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ agar sesuai dengan analisis kebutuhan.
5. Mengukur usabilitas pada sistem informasi pasien operasi dan visite untuk tenaga medis Rumah Sakit XYZ pada perangkat bergerak berbasis Android.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah tenaga medis di Rumah Sakit XYZ dapat mengakses data pasien operasi dan visite dengan mudah dan fleksibel, sehingga mengurangi kesalahan yang dapat terjadi karena misinformasi data pasien operasi dan mengurangi waktu yang dibutuhkan untuk tenaga medis dokter dalam mendapatkan data pasien operasi dan visite. Selain itu, agar rumah sakit dengan proses bisnis yang sama dengan Rumah Sakit XYZ mendapatkan gambaran tentang metode yang dapat diterapkan untuk mempercepat alur informasi bagi tenaga medis untuk mengakses data pasien, visite, dan operasi.

1.5 Batasan masalah

1. Aplikasi yang dibangun menggunakan struktur basis data Rumah Sakit XYZ.
2. Data pasien yang dapat diakses oleh tenaga medis adalah pasien rawat inap di Rumah Sakit XYZ.

1.6 Sistematika pembahasan

BAB I Pendahuluan:

Bab pendahuluan berisi latar belakang yang menjelaskan secara sederhana permasalahan yang melatarbelakangi penelitian ini dan solusi yang dilakukan/dibangun untuk menyelesaikan masalah. Memuat rumusan masalah berupa pertanyaan untuk dijadikan landasan permasalahan yang akan diselesaikan. Memuat tujuan dan manfaat penelitian serta batasan dari penelitian dan sistematika penulisan.

BAB II Landasan Kepustakaan:

Bab ini memuat landasan teori dan penelitian yang mendukung dan relevan terhadap permasalahan yang dihadapi peneliti. Memuat penjelasan dasar mengenai Rumah Sakit XYZ dan proses bisnis yang berjalan, sistem operasi Android, teknologi yang digunakan dalam pengimplementasian sistem, dan penelitian sebelumnya yang memiliki hubungan dengan pembangunan sistem informasi berbasis *mobile* untuk rumah sakit.

BAB III Metodologi:

Bab ini memuat alur proses yang dilakukan dalam penelitian. Penjelasan mengenai tiap proses dalam metodologi pembangunan aplikasi dari studi kepustakaan hingga pengambilan kesimpulan.

BAB IV Analisis dan Perancangan:

Bab ini memuat analisis kebutuhan aplikasi yang digambarkan menggunakan *use case diagram* dan *activity diagram*. Memuat perancangan aplikasi menggunakan *sequence diagram*, *entity relationship diagram*, dan *mockup interface* sebagai acuan untuk proses implementasi aplikasi.

BAB V Implementasi:

Bab ini berisi spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam proses implementasi. Memuat implementasi basis data yang menjelaskan atribut dan fungsi masing-masing tabel dalam basis data yang digunakan. Memuat implementasi kelas yang menjelaskan isi dan hubungan antar kelas dalam implementasi sistem. Terdapat penjelasan tentang implementasi kode dalam bentuk pseudocode untuk fungsi-fungsi penting dalam aplikasi yang digunakan dalam proses implementasi. Memuat implementasi antar muka berdasarkan hasil perancangan.

BAB VI Pengujian dan Analisis:

Bab ini membahas tentang proses pengujian aplikasi dan analisis hasil pengujian tersebut. Memuat pengujian *blackbox* untuk kebutuhan fungsional, dan pengujian SUS(*System Usability Scale*) untuk tingkat *usability* aplikasi berdasarkan jawaban atas pertanyaan yang diberikan kepada responden pengujian.

BAB VII Penutup:

Bab ini memuat kesimpulan dari hasil penelitian yang dilakukan dan saran untuk penelitian selanjutnya dengan topik yang sama dan saran dalam pengembangan aplikasi selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Terdapat penelitian serupa berjudul “Sistem Informasi Visite Dokter Rawat Inap Pada Rumah Sakit Umum Islam Harapan Anda” oleh Salis Farkhanani Lathifah dan Edi Sugiarto yang dilaksanakan untuk mengoptimalkan pelaksanaan proses visite dokter yang memiliki jadwal praktik di tempat yang terpisah dengan rumah sakit. Sistem informasi yang dibangun dalam penelitian Salis & Edi menggunakan basis *web application*, dan hasil yang didapatkan adalah sebuah sistem yang mempermudah pengecekan jadwal visite dokter, penyimpanan data riwayat visite dokter, dan penyimpanan data riwayat pasien.

Berbeda dengan penelitian Salis & Edi yang bertujuan untuk kemudahan proses penyimpanan data dan arsip, penelitian ini lebih mengutamakan faktor kecepatan pengambilan informasi dari basis data yang sudah ada kepada dokter yang bertanggung jawab untuk pelaksanaan visite dan operasi, selain itu *platform* yang digunakan untuk pembangunan aplikasi adalah berbasis *mobile*. Akan tetapi proses pembangunan aplikasi mulai dari tahap analisis hingga pengujian masih memiliki kesamaan proses, sehingga dapat dijadikan sebagai bahan pengkajian pustaka.

2.2 Model *Waterfall*

Salah satu metode pengembangan perangkat lunak adalah dengan menggunakan model *waterfall*. Model *waterfall* merupakan proses berdasarkan perencanaan dimana seluruh aktivitas harus dicanangkan dan dijadwalkan sebelum mulai dikerjakan (Sommerville, 2011). Pada dasarnya, prinsip dasar *waterfall* memiliki kesamaan fundamental dengan aktivitas pengembangan perangkat lunak.

Terdapat 4 tahap utama dalam pengembangan perangkat lunak dengan model *waterfall*, yaitu:

1. Analisis Kebutuhan: Penentuan batasan dan tujuan pengembangan dengan calon pengguna. Dijelaskan secara detil dan berfungsi sebagai tahap penentuan spesifikasi sistem.
2. Perancangan: Pencanaan arsitektur sistem baik secara perangkat lunak maupun perangkat keras melalui pengalokasian sumber daya. Dalam proses perencanaan, ditentukan abstraksi sistem dan relasi dalam sistem.
3. Implementasi dan pengujian unit: Realisasi program dalam bentuk program atau unit program. Setiap unit program diuji untuk melihat kesesuaian dengan spesifikasi yang ditentukan.
4. Integrasi dan pengujian sistem: Pengintegrasian setiap unit program menjadi satu kesatuan untuk memastikan kebutuhan perangkat

lunak sudah terpenuhi. Setelah tahap integrasi, perangkat lunak diserahkan kepada pelanggan.

5. Operasi dan perawatan: Penggunaan perangkat lunak di lingkungan sebenarnya. Dalam proses perawatan dilakukan pembenahan kesalahan yang tidak terdeteksi di tahap pengujian, meningkatkan implementasi unit sistem, dan menambah layanan sistem jika terdapat kebutuhan baru.

2.3 Diagram Kelas (*Class Diagram*)

Berdasarkan pemaparan Sommerville (2011) diagram kelas adalah diagram yang digunakan untuk mengembangkan model sistem berbasis objek, untuk menunjukkan hubungan antar kelas yang ada pada sistem tersebut. Hubungan antar kelas dapat memiliki arti bahwa sebuah kelas dapat mengetahui informasi dari kelas lain yang saling berhubungan.

Dalam pengembangan model kelas diagram, sebuah objek merepresentasikan sesuatu di dunia nyata. Walaupun terkadang objek juga dapat berbentuk sistem yang semu/maya. Dan dalam tahap implementasi objek-objek lain ditambahkan untuk menyediakan fungsionalitas bagi sistem yang dibangun.

Penggambaran diagram kelas dalam UML(*Unified Modeling Language*) dapat dilakukan dalam beragam tingkat kedalaman. Tahapan yang dilakukan dalam pembuatan model pada dasarnya adalah: observasi lingkungan, identifikasi objek penting, dan pengubahan hal-hal tersebut ke dalam bentuk kelas. Penggambaran kelas-kelas tersebut dapat dilakukan dengan sederhana melalui sebuah kotak yang mendefinisikan setiap objek dan atribut-atributnya.

2.4 Sistem Operasi Android

Android adalah sistem operasi ciptaan perusahaan teknologi Google, yang dikembangkan dari kernel Linux untuk digunakan pada perangkat *mobile* dengan layar sentuh seperti *smartphone* dan perangkat *tablet*.

2.4.1 Komponen Aplikasi Android

Menurut Zigurd et al.(2011), terdapat 4 komponen utama dalam aplikasi Android, yaitu:

1. *Activities*: Bagian-bagian dari kode yang dapat dieksekusi, yang diinstansiasi oleh pengguna atau sistem operasi, dan berjalan selama dibutuhkan.
2. *Services*: Bagian kode yang dapat dieksekusi yang berjalan di latar belakang sistem operasi sejak awal perangkat dihidupkan, hingga perangkat dimatikan. *Services* tidak memiliki tampilan yang dapat diakses secara langsung oleh pengguna.
3. *Broadcast and Intent Receivers*: Yaitu bagian pada aplikasi yang berfungsi untuk menerima dan merespon permintaan dari proses lain yang berjalan. Proses yang melakukan permintaan dapat berasal

dari perangkat Android itu sendiri, atau berasal dari aplikasi lain yang berjalan di sistem. Melalui *intent receiver*, sebuah *activity* dapat mengeksekusi kode/perintah pada aplikasi lain.

4. *Content Providers*: Yaitu bagian pada aplikasi yang berfungsi untuk berbagi data antar aplikasi. Penyedia konten menggunakan URI (*Unified Resource Identifier*) untuk menyelesaikan permintaan dari aplikasi lain yang tidak memiliki informasi mengenai penyedia konten yang diakses.

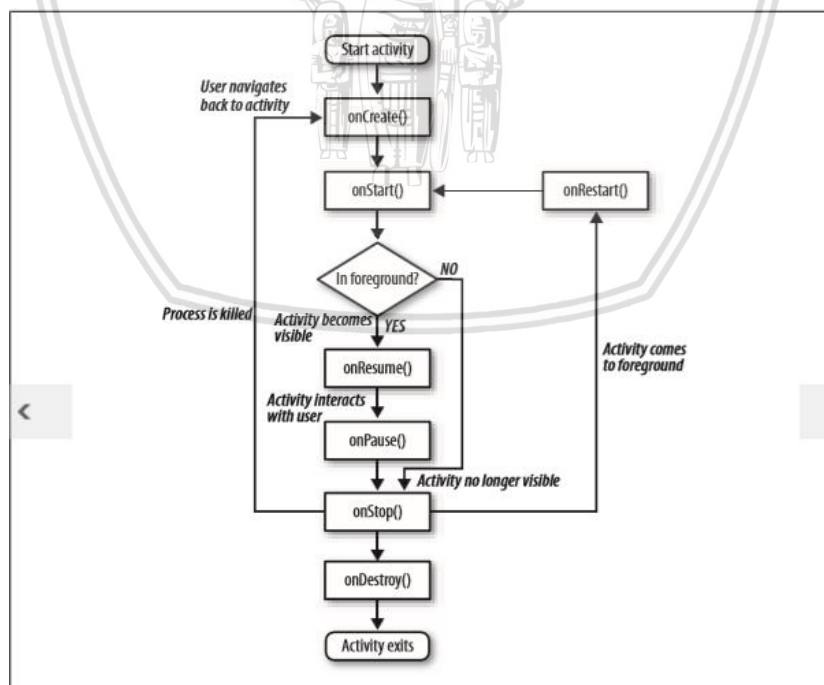
Sebuah aplikasi tidak harus menggunakan seluruh komponen android, akan tetapi sebuah kode dalam aplikasi android yang baik akan menggunakan seluruh komponen yang tersedia.

2.4.2 Daur Hidup *Activity* pada Aplikasi Android

Menurut Zigurd et al.(2011) dalam sistem operasi Android, setiap aplikasi harus mempertimbangkan kapabilitas perangkat *mobile* yang terbatas pada sumber daya memori dan baterai, sehingga terdapat mekanisme dalam Android yang berfungsi untuk menghemat penggunaan sumberdaya-sumberdaya yang ada. Mekanisme-mekanisme ini tertuang dalam bentuk daur hidup *activity* Android seperti yang tercantum pada Gambar 2.1.

Ada beberapa proses dalam daur hidup *activity* Android, yaitu:

1. *onCreate*: Yaitu proses yang dieksekusi ketika sebuah *activity* pertama kali dibuat. Pada bagian ini bagian tampilan dan alur data diciptakan.



Gambar 2.1 Daur Hidup *Activity* pada Aplikasi Android

2. *onCreate*: Yaitu proses yang dieksekusi ketika sebuah *activity* pertama kali dibuat. Pada bagian ini bagian tampilan dan alur data diciptakan.
3. *onStart*: Yaitu bagian yang dieksekusi ketika sebuah *activity* pertama kali ditampilkan di layar perangkat.
4. *onResume*: Dieksekusi setelah proses *onStart* jika proses yang dilakukan berada di latarbelakang aplikasi ketika dieksekusi. Proses *onResume* juga dieksekusi jika aplikasi dibuka kembali setelah tertutup oleh aplikasi lain dalam proses *onPause*.
5. *onPause*: Yaitu bagian aplikasi yang dieksekusi jika perangkat mengakses *activity* lain dengan memanggil proses *onResume*. Ketika proses *onPause* dieksekusi, aplikasi tidak memiliki akses untuk ditampilkan pada layar perangkat.
6. *onStop*: Dieksekusi jika sebuah *activity* sudah tidak bisa dilihat kembali, yang disebabkan oleh pengambilalihan proses oleh aplikasi lain, atau dieksekusinya proses *onDestroy*.
7. *onDestroy*: Bagian terakhir dimana aplikasi dapat mengeksekusi proses sebelum *activity* dimusnahkan. Ketika aplikasi dimusnahkan maka aktivitas yang ada pada aplikasi tersebut sudah berakhir. Proses *onDestroy* dapat terjadi karena dilakukan secara sengaja oleh pengguna atau dilakukan secara otomatis oleh sistem operasi untuk membebaskan memori perangkat.

2.4.3 Daur Hidup *Service* pada Aplikasi Android

Services pada Android memiliki daur hidup yang mirip dengan *activities*, yang menurut Zigurd et al.(2011) memiliki perbedaan pada bagian:

1. *onResume*, *onStart*, dan *onPause* tidak digunakan: Karena sebuah *service* tidak memiliki tampilan/*view*, maka proses *onResume*, *onStart*, dan *onPause* tidak diperlukan dalam daur hidup *service*.
2. *onBind*: Jika dalam sebuah aplikasi dibutuhkan sambungan terus-menerus ke sebuah *service*, maka dapat digunakan proses *onBind*.
3. *onDestroy*: Sama seperti *activity*, sebuah *service* juga memiliki proses *onDestroy* jika *service* tersebut sudah tidak lagi digunakan. Android akan memanggil proses *onDestroy* jika tidak ada lagi sambungan ke *service*. Judul tabel harus berada dalam satu halaman dengan tabelnya. Fitur yang relevan dalam program pengolah kata dapat digunakan untuk menjaga konsistensi ini.

2.5 System Usability Scale (SUS)

SUS adalah metode pengujian usability yang cepat dan dapat diandalkan (Brooke, 1996). Dalam metode SUS, responden diberikan 10 pertanyaan yang memberikan pandangan umum tentang usability dari sebuah

sistem. Responden dapat menjawab tiap pertanyaan dalam 5 bobot, mulai dari sangat tidak setuju, tidak setuju, netral, setuju, dan sangat setuju.

Jenis pertanyaan pada metode SUS berselang-seling antara pertanyaan negatif dengan pertanyaan positif, yang tiap jenis pertanyaan memiliki kalkulasi skor tersendiri. Untuk setiap pertanyaan positif, cara perhitungan dilakukan dengan mengurangi bobot jawaban dengan 1, yaitu $(x-1)$ dimana x merupakan bobot jawaban dengan pertanyaan positif. Maka jika responden menjawab pertanyaan dengan bobot sangat setuju (bobot 5), maka skor SUS untuk pertanyaan tersebut adalah 4. Sedangkan untuk pertanyaan negatif, bobot dari jawaban akan menjadi pengurang 5, yaitu $(5-y)$ dimana y adalah bobot jawaban untuk pertanyaan negatif. Maka jika responden menjawab sangat tidak setuju (bobot 1) pada pertanyaan negatif, skor SUS untuk pertanyaan tersebut adalah 4.

Total skor untuk masing-masing pertanyaan kemudian dijumlahkan dan dikali dengan 2,5 untuk mendapatkan skor total SUS tiap responden. Skor SUS akan berada dalam rentang 0-100 dengan kriteria *Not Acceptable*, *Marginal*, dan *Acceptable* masing-masing untuk skor 0-50.9, 51-70.9, dan 71-100 (Ardyansyah, 2016). Penilaian total skor SUS untuk seluruh responden dapat digambarkan dalam persamaan berikut: $\frac{\sum(x-1) + \sum(5-y) \times 2.5}{n}$ dimana x merupakan bobot jawaban untuk pertanyaan positif, y merupakan bobot jawaban untuk pertanyaan negatif, dan n merupakan total responden.

BAB 3 METODOLOGI

3.1 Tipe dan Metode Penelitian

Tipe penelitian ini adalah implementatif pengembangan, bertujuan untuk membuat sebuah perangkat lunak dengan menerapkan prinsip-prinsip rekayasa yang meliputi analisis, perancangan, implementasi, dan pengujian dengan menggunakan metode *waterfall*.

3.2 Lokasi Penelitian

Penelitian ini dilakukan di Rumah Sakit XYZ.

3.3 Metode Pengumpulan data

Dalam pelaksanaan penelitian ini terdapat 2 metode pengumpulan data yaitu:

- Studi kepustakaan: pengumpulan data berdasarkan penelitian sebelumnya dengan kesamaan topik, tipe penelitian, atau metode. Data relevan yang berasal dari literatur, yang mendukung satu atau lebih proses dalam penelitian.
- Studi lapangan: melakukan wawancara dengan *stakeholder* Rumah Sakit XYZ dan pihak departemen IT Rumah Sakit XYZ untuk proses identifikasi permasalahan, analisis kebutuhan, dan pengujian aplikasi.

3.4 Sistematika Penelitian

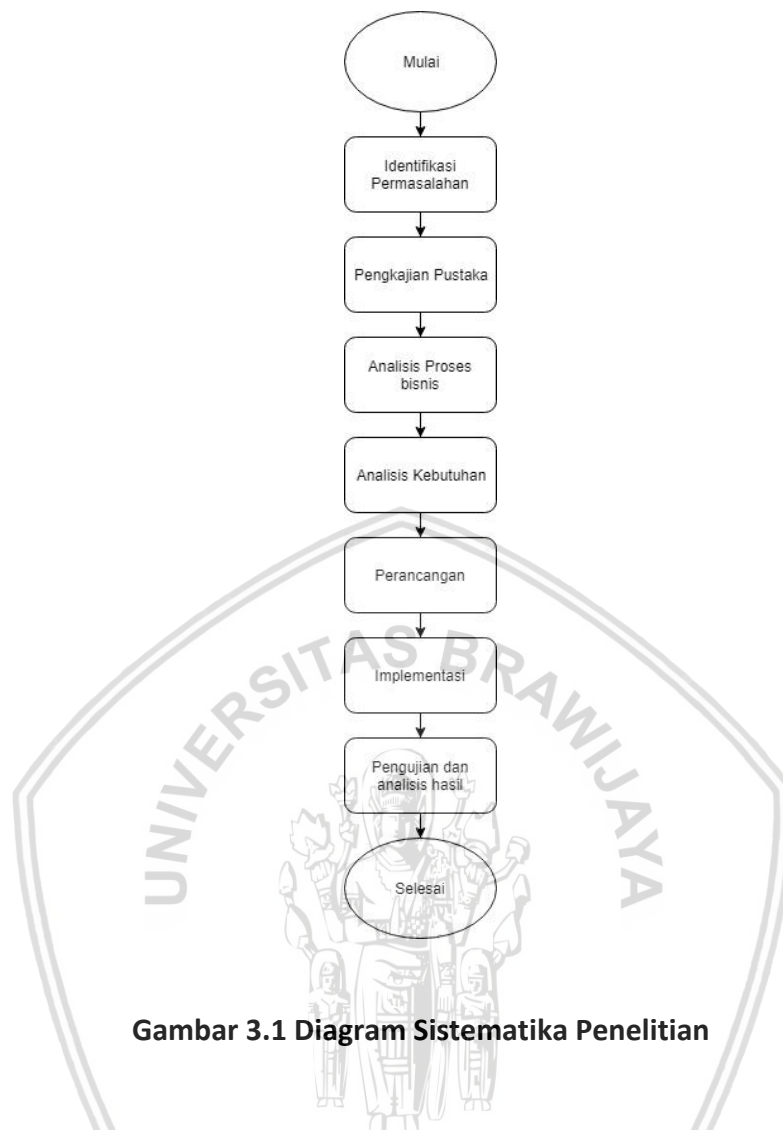
Gambar 3.1 menjelaskan deretan aktivitas yang secara sistematis akan dilaksanakan dalam penelitian ini.

3.4.1 Identifikasi Permasalahan

Melakukan wawancara dengan pihak Rumah Sakit XYZ terkait permasalahan yang dapat diselesaikan dengan dibangunnya perangkat lunak. Pencarian data yang melatarbelakangi masalah yang terjadi. Diskusi dengan *stakeholder* Rumah Sakit XYZ mengenai potensi solusi yang dapat dilakukan dan menentukan tujuan dan manfaat solusi tersebut bagi *stakeholder* dan penelitian ini. Menentukan batasan masalah untuk solusi permasalahan.

3.4.2 Pengkajian Pustaka

Melakukan pengkajian literatur dan penelitian sebelumnya, yang memiliki dasar untuk membantu penelitian dan proses yang dilakukan dalam penelitian. Mengkaji dasar teori dari solusi yang digunakan dalam penelitian, dalam hal ini pengembangan aplikasi *mobile* berbasis Android dari literatur yang tersedia.



Gambar 3.1 Diagram Sistematika Penelitian

3.4.3 Analisis Proses Bisnis

Menganalisa proses bisnis yang berjalan di Rumah Sakit XYZ yang berhubungan dengan kegiatan pengambilan data pasien visite dan operasi dengan cara observasi langsung, wawancara, dan dokumentasi resmi alur proses Rumah Sakit XYZ. Melakukan pemodelan untuk proses bisnis yang sedang berjalan yang akan dijadikan landasan untuk perbandingan proses sebelum implementasi sistem aplikasi mobile dengan setelah implementasi sistem aplikasi mobile.

3.4.4 Analisis Kebutuhan

Melakukan wawancara dengan *stakeholder* Rumah Sakit XYZ, untuk mengidentifikasi kebutuhan aplikasi yang akan dibangun. Mendefinisikan kebutuhan fungsional dan non-fungsional berdasarkan pertimbangan *stakeholder* Rumah Sakit XYZ. Membuat diagram *use case* untuk menjelaskan kebutuhan utama aplikasi yang dibangun berdasarkan aktor.

3.4.5 Perancangan

Membuat diagram kelas untuk dijadikan landasan pengimplementasian aplikasi. Membuat diagram *sequence* untuk menjelaskan hubungan antar kelas yang berinteraksi dalam aplikasi yang dibangun. Membuat rancangan antarmuka aplikasi sebagai acuan implementasi. Membuat perancangan basis data sebagai landasan implementasi.

3.4.6 Implementasi

Mendefinisikan spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam proses implementasi aplikasi. Pengimplementasian basis data berdasarkan hasil perancangan. Pengimplementasian kelas beserta relasi-relasi antar kelas. Pengimplementasian antarmuka aplikasi berdasarkan perancangan. Penjelasan implementasi kode dalam bentuk *pseudocode* untuk fungsi-fungsi penting pada aplikasi.

3.4.7 Pengujian dan Analisis Hasil

Pengujian *black box* dilakukan untuk melihat kesesuaian kebutuhan fungsional aplikasi. Pengujian usabilitas dengan metode SUS(*System Usability Scale*) dilakukan untuk mendapatkan skor penerimaan aplikasi terhadap calon pengguna. Melakukan analisis validitas hasil uji *blackbox* dan hasil uji SUS.

3.4.8 Kesimpulan Hasil dan Saran

Menyimpulkan hasil penelitian berdasarkan pengujian yang dilakukan dan saran untuk penelitian selanjutnya yang memiliki kesamaan topik, metode, atau tipe penelitian.

BAB 4 ANALISIS DAN PERANCANGAN

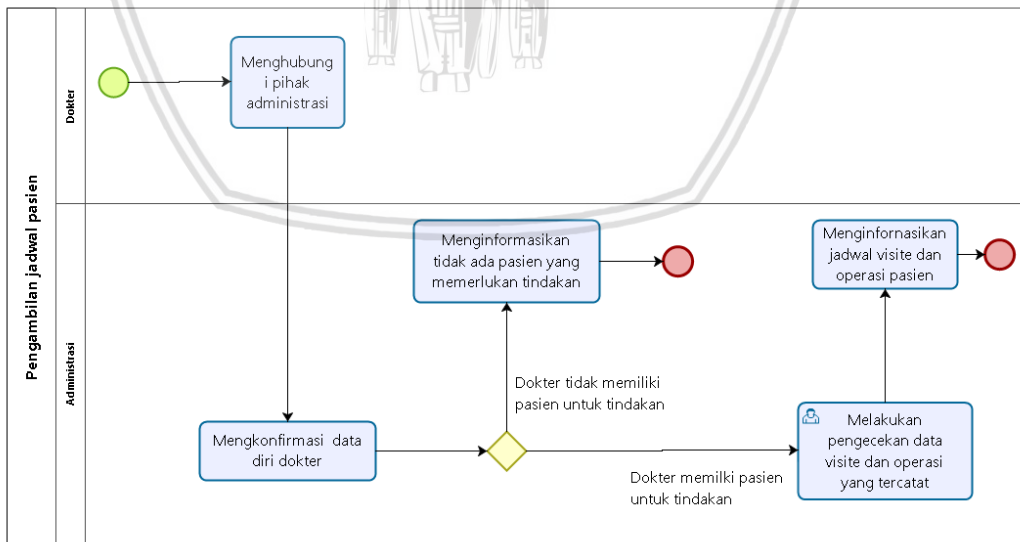
4.1 Analisis Kebutuhan

Pada bagian ini akan dijelaskan proses-proses yang dilakukan dalam menganalisis kebutuhan untuk aplikasi yang akan dibangun. Meliputi analisis kebutuhan sistem, gambaran dasar sistem, analisis kebutuhan fungsional diagram *use case* dan pembuatan *use case scenario*.

4.1.1 Analisis Kebutuhan Sistem

Sistem yang akan dibangun adalah aplikasi berbasis *mobile* yang memanfaatkan infrastruktur teknologi informasi yang sudah ada di Rumah Sakit XYZ. Saat ini basis data untuk menyimpan data pasien sudah dimiliki dan digunakan oleh Rumah Sakit XYZ. Oleh karena itu kebutuhan utama untuk sistem yang dibangun yaitu:

1. Mengambil informasi pasien visite (rawat inap) dari basis data Rumah Sakit XYZ berdasarkan dokter yang mengakses, untuk menampilkan informasi mendasar seperti ruangan rawat inap, penyakit yang diderita, lama waktu rawat, dan lain sebagainya.
2. Mengambil informasi jadwal operasi dari basis data Rumah Sakit XYZ berdasarkan dokter yang mengakses, untuk menampilkan data operasi seperti ruangan operasi, waktu operasi, dan pasien yang dioperasi.

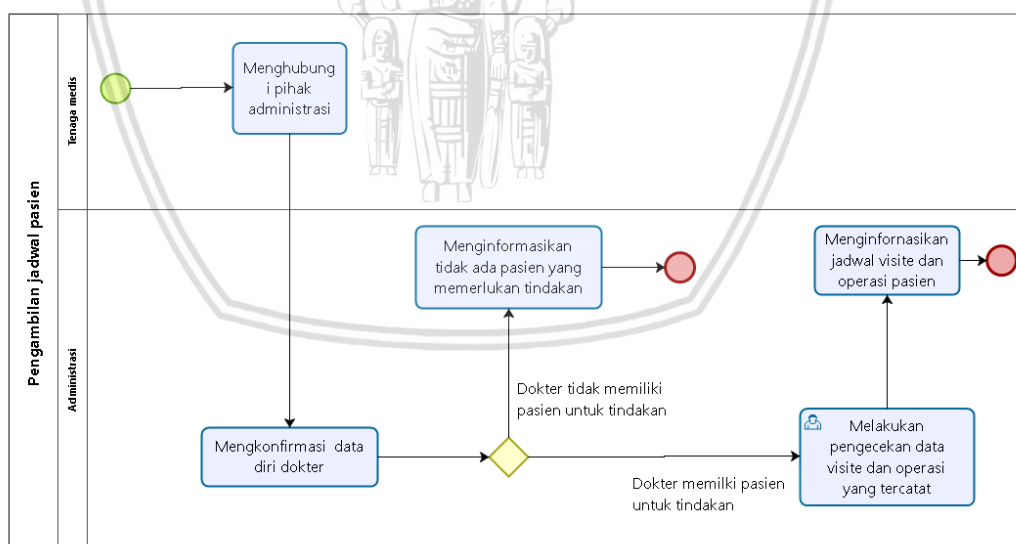


Gambar 4.1 Alur proses bisnis informasi pasien

Seperti yang tercantum pada Gambar 4.1 Terdapat 2 aktor utama dalam proses pengambilan informasi pasien di Rumah Sakit XYZ, yaitu dokter dan departemen administrasi. Dokter adalah pihak yang akan melaksanakan operasi atau visite kepada pasien, dan bidang administrasi adalah pusat penyimpanan data pasien di Rumah Sakit XYZ. Ketika dokter ingin mendapatkan data pasien, maka dokter menghubungi pihak administrasi dengan menyertakan data diri dokter untuk dikonfirmasi. Pihak administrasi kemudian melakukan pengecekan data pasien yang memerlukan tindakan dari dokter yang bersangkutan. Jika ditemukan, maka pihak administrasi akan menginformasikan kembali kepada dokter mengenai data pasien, jadwal visite dan operasi, dan ruang visite(rawat inap) atau ruang operasi untuk melakukan tindakan.

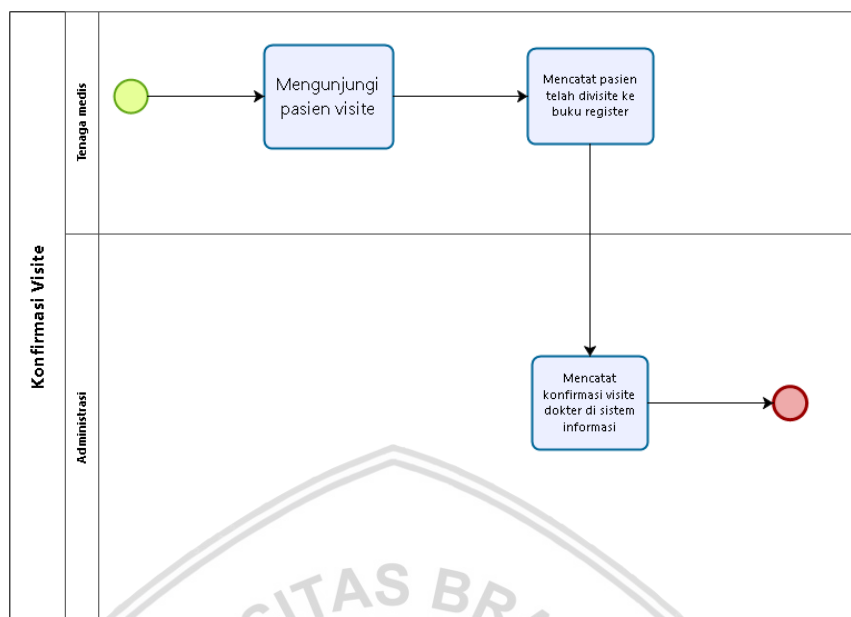
Proses menghubungi pihak administrasi oleh dokter dapat dilakukan secara langsung maupun melalui perangkat telepon. Pihak administrasi juga dapat memberikan informasi pasien secara independen tanpa perlu permintaan dokter melalui pengiriman pesan singkat.

Berdasarkan hasil analisis proses bisnis, untuk meminimalisir waktu yang digunakan dalam mendapatkan informasi, maka peran aktor administrasi harus dikurangi. Oleh karena itu dalam perancangan proses bisnis dengan implementasi sistem, peran administrasi diganti dengan aktor baru yaitu sistem, dimana proses penerimaan dan pemrosesan data dari aktor tenaga medis akan langsung direspon secara langsung oleh sistem yang terimplementasi, seperti yang tergambar pada Gambar 4.2.



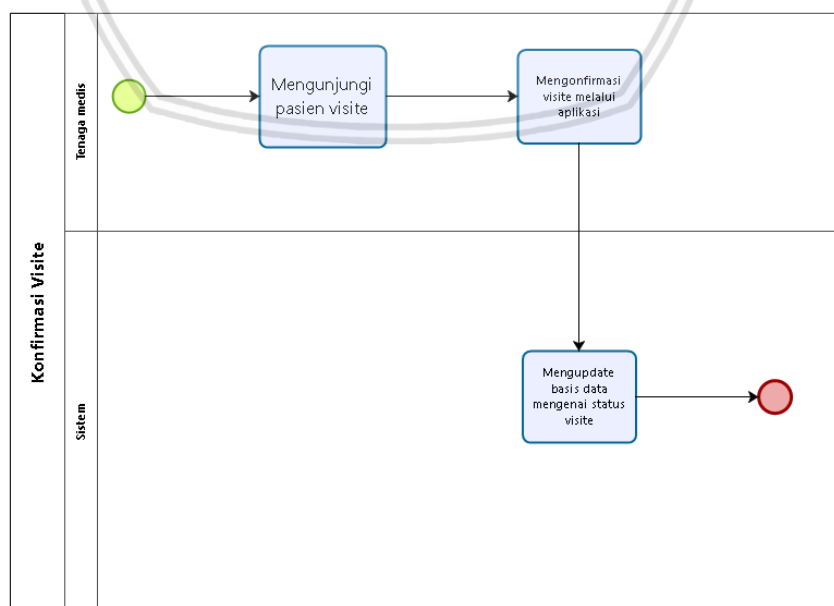
Gambar 4.2 Alur proses bisnis dengan sistem

Selain kebutuhan utama untuk mendapatkan informasi pasien visite dan operasi, dalam proses analisis juga terdapat kebutuhan sampingan, yaitu mempermudah konfirmasi visite dan perbaharuan status operasi secara langsung oleh tenaga medis. Pada saat ini, proses konfirmasi dilakukan sesuai proses yang tergambar pada Gambar 4.3, dan proses operasi pada Gambar 4.5.

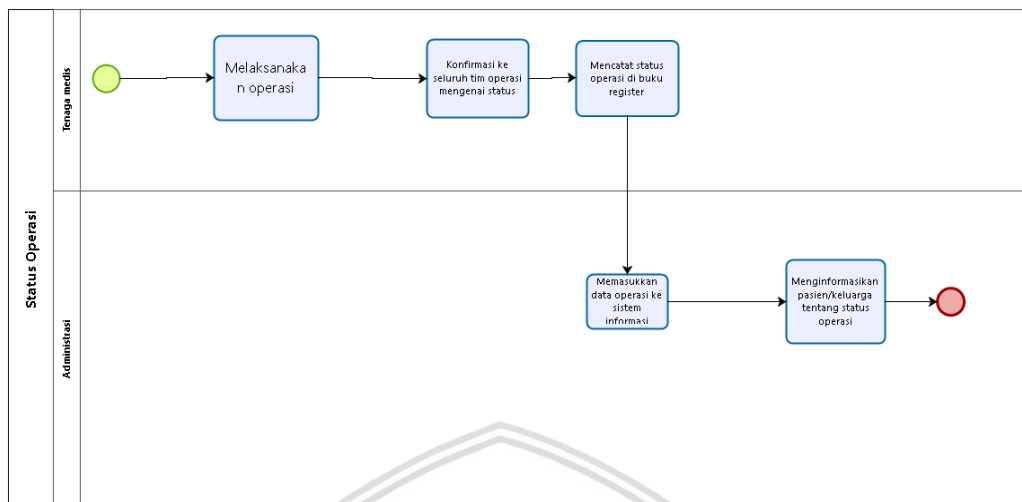


Gambar 4.3 Alur proses bisnis konfirmasi visite

Pada proses bisnis konfirmasi visite, tenaga medis mencatat pasien yang telah divisite melalui buku register, lalu staff administrasi/rekam medik kemudian memasukkan data yang ditulis oleh tenaga medis kedalam sistem informasi rumah sakit. Dari proses ini terlihat proses yang redundan, yaitu pencatatan data visite oleh tenaga medis dan oleh staff rekam medis. Oleh karena itu dalam perancangan konfirmasi visite dengan sistem, peran staff digantikan oleh sistem, sehingga tenaga medis dapat secara langsung mengubah status visite pasien tanpa perlu melalui staff administrasi. Proses konfirmasi visite dengan sistem tercantum pada Gambar 4.4

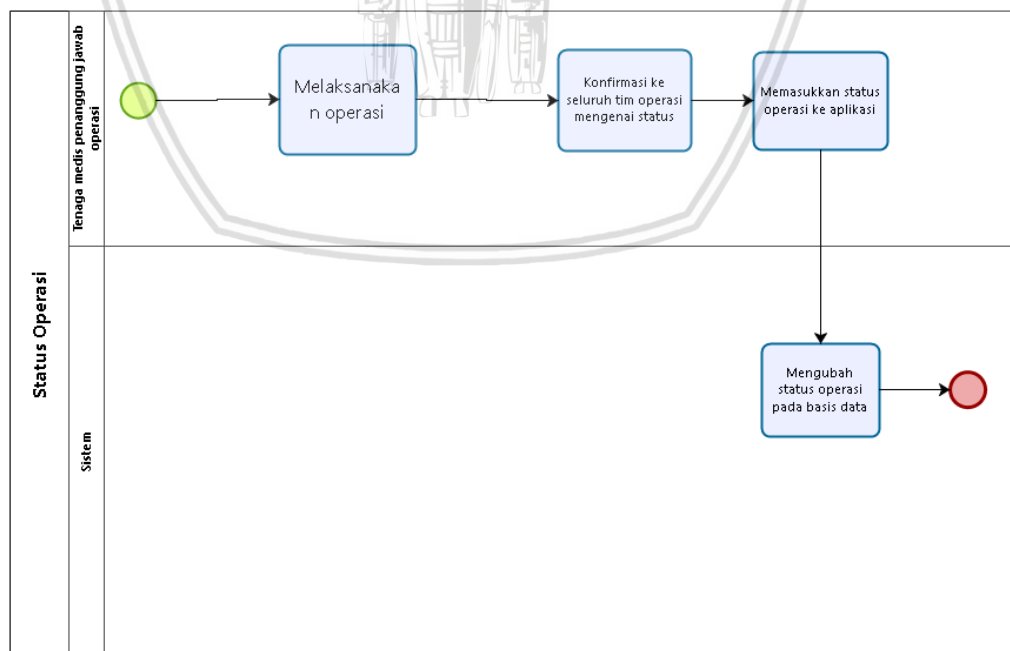


Gambar 4.4 Alur proses bisnis konfirmasi visite dengan sistem



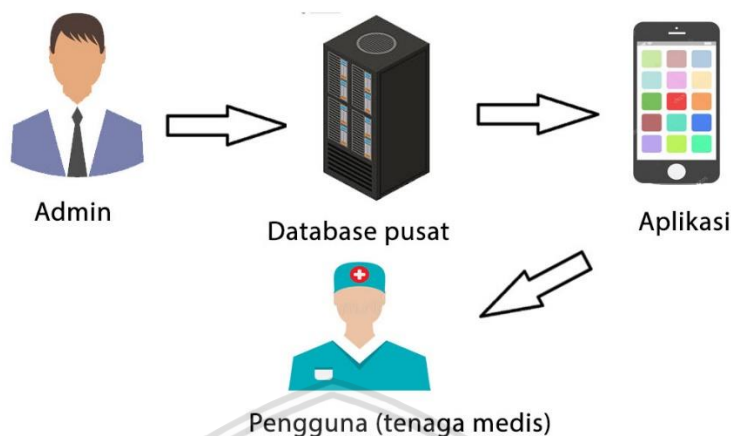
Gambar 4.5 Alur proses bisnis status operasi

Untuk proses status operasi, kepala tim operasi juga tetap harus mencatat informasi operasi melalui buku register, dan kemudian pihak staff rumah sakit memasukkan data status operasi kedalam sistem informasi. Sama seperti proses konfirmasi visite, untuk mengurangi redundansi aktivitas, maka perancangan proses untuk status operasi dengan aplikasi menggunakan sistem sebagai pengganti staff rekam medis, untuk mempermudah dan mempercepat pengubahan status operasi di basis data rumah sakit. Rancangan proses setelah implementasi sistem seperti pada Gambar 4.6



Gambar 4.6 Alur proses bisnis status operasi dengan sistem

4.1.2 Gambaran Sistem



Gambar 4.3 Gambaran Sistem

Seperti yang terlihat pada Gambar 4.3, aplikasi yang dibangun akan menggunakan data dari *database* yang telah dibangun di Rumah Sakit XYZ. Dalam *database* ini terdapat data pasien, dokter, dan jadwal ruang operasi dan visite, yang berasal dari proses administrasi yang dilakukan oleh *admin* sistem sehingga untuk pengembangan aplikasi akan dilakukan integrasi antara basis data Rumah Sakit XYZ dengan sistem yang akan dibangun, untuk kemudian ditampilkan di aplikasi dan digunakan oleh tenaga medis dokter.

4.1.3 Analisis Kebutuhan Fungsional

Dalam pembangunan aplikasi ini kebutuhan fungsional akan berfokus pada aktor utama saja yaitu pengguna aplikasi (tenaga medis dokter). Daftar kebutuhan fungsional seperti yang tercantum pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

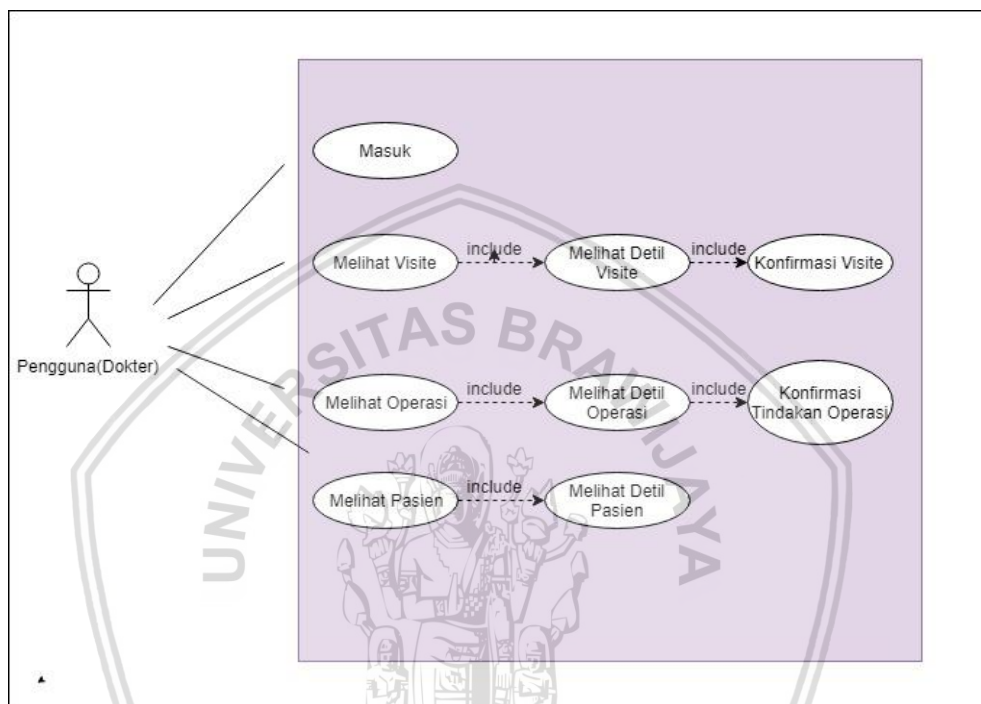
Kode Fungsional	Nama Fungsional	Deskripsi
KF-01	Masuk	Fungsi agar pengguna yang telah terdaftar sebelumnya dapat masuk untuk mendapatkan akses ke aplikasi.
KF-02	Melihat daftar pasien	Fungsi untuk melihat daftar pasien yang ditangani oleh pengguna.
KF-03	Melihat detail pasien	Fungsi untuk melihat informasi terkait pasien

		tertentu. Informasi berupa penyakit, waktu mulai rawat inap, ruang rawat dan jadwal penanganan.
KF-04	Melihat halaman visite	Fungsi untuk melihat daftar visite pengguna. Diurutkan berdasarkan tanggal dan waktu.
KF-05	Melihat detil visite	Fungsi untuk mendapatkan informasi mengenai visite pada waktu tertentu. Memuat ruang visite dan data pasien yang di visit.
KF-06	Melihat halaman operasi	Fungsi untuk melihat daftar tindakan operasi yang akan dilakukan pengguna. Diurutkan berdasarkan tanggal dan waktu.
KF-07	Melihat detil operasi	Fungsi untuk mendapatkan informasi mengenai tindakan operasi pada waktu tertentu. Memuat ruang operasi, tindakan yang dilakukan, dan pasien yang akan ditindak.
KF-08	Konfirmasi visite	Fungsi untuk mengonfirmasi bahwa pengguna(dokter) telah melaksanakan visite ke pasien tertentu.
KF-09	Konfirmasi status operasi	Fungsi untuk mengonfirmasi status operasi menjadi selesai atau diekstensi(tindakan belum selesai dan dilaksanakan dilain waktu).

4.1.4 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional pada sistem yang akan dibangun adalah usabilitas. Usabilitas yang dibutuhkan adalah kecepatan dan kemudahan pengguna dalam menggunakan aplikasi.

4.1.5 Use Case



Gambar 4.4 Diagram use case

Dalam rancangan aplikasi yang dibangun dalam diagram use case pada Gambar 4.4, terdapat 2 buah fungsi utama yaitu melihat halaman visite dan melihat halaman operasi. Masing-masing halaman memiliki sebuah fungsi lain untuk melihat detil, sehingga terdapat relasi *include*.

4.1.6 Use Case Scenario

Dalam tahap ini dilakukan pembuatan skenario untuk masing-masing *use case*. Dimulai dari Tabel 4.2 untuk skenario masuk, Tabel 4.3 untuk skenario daftar pasien, Tabel 4.4 untuk skenario detil pasien, Tabel 4.5 untuk skenario jadwal visite, Tabel 4.6 untuk skenario detil visite, Tabel 4.7 untuk skenario jadwal operasi, Tabel 4.8 untuk skenario detil operasi, Tabel 4.9 untuk skenario konfirmasi visite, dan Tabel 4.10 untuk skenario update status operasi.

Tabel 4.2 Skenario masuk

Masuk

Aktor	Pengguna
Tujuan	Untuk berhasil masuk kedalam sistem.
Kondisi Sukses	Pengguna berhasil masuk kedalam sistem(dapat mengakses halaman utama).
Kondisi Gagal	Pengguna tidak dapat masuk kedalam sistem.
Kondisi Awal	Pengguna membuka aplikasi.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna mengisi <i>username</i> dan <i>password</i> yang telah terdaftar. 2. Pengguna mengkonfirmasi <i>username</i> dan <i>password</i> yang dimasukkan. 3. Sistem menampilkan halaman utama.
Alir Alternatif 1	<ol style="list-style-type: none"> 1. Pengguna mengisi username yang tidak terdaftar dan memasukkan password. 2. Sistem memberitahu bahwa username dan password tidak tepat.
Alir Alternatif 2	<ol style="list-style-type: none"> 1. Pengguna mengisi username yang terdaftar dan memasukkan password yang salah. 2. Sistem memberitahu bahwa username dan password tidak tepat.

Tabel 4.3 Skenario melihat daftar pasien

Melihat daftar pasien	
Aktor	Pengguna.
Tujuan	Untuk mengakses daftar pasien pengguna.
Kondisi Sukses	Pengguna berhasil mengakses daftar pasien.
Kondisi Gagal	Pengguna tidak dapat mengakses daftar pasien.

Kondisi Awal	Pengguna telah masuk kedalam aplikasi dan berada di halaman awal.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan daftar pasien. 2. Sistem menampilkan halaman daftar pasien.

Tabel 4.4 Skenario melihat detil pasien

Melihat detil pasien	
Aktor	Pengguna.
Tujuan	Untuk mengakses detil individu pasien pengguna.
Kondisi Sukses	Pengguna berhasil mengakses detil pasien yang dipilih.
Kondisi Gagal	Pengguna tidak dapat mengakses detil pasien yang dipilih.
Kondisi Awal	Pengguna telah memilih pilihan daftar pasien pada halaman awal.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar pasien yang ada. 2. Sistem menampilkan halaman detil pasien yang dipilih pengguna.

Tabel 4.5 Skenario melihat jadwal visite

Melihat jadwal visite	
Aktor	Pengguna.
Tujuan	Untuk mengakses jadwal visite pengguna.
Kondisi Sukses	Pengguna berhasil mengakses jadwal visite.
Kondisi Gagal	Pengguna tidak dapat mengakses jadwal visite.
Kondisi Awal	Pengguna telah masuk kedalam aplikasi dan berada di halaman awal.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan jadwal visite.

	2. Sistem menampilkan halaman jadwal visite pengguna.
--	---

Tabel 4.6 Skenario melihat detail visite

Melihat detail visite	
Aktor	Pengguna.
Tujuan	Untuk mengakses detail visite pengguna.
Kondisi Sukses	Pengguna berhasil mengakses detail visite yang dipilih.
Kondisi Gagal	Pengguna tidak dapat mengakses detail visite yang dipilih.
Kondisi Awal	Pengguna telah memilih pilihan jadwal pasien pada halaman utama.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar visite yang ada. 2. Sistem menampilkan halaman detail visite yang dipilih pengguna.

Tabel 4.7 Skenario melihat jadwal operasi

Melihat jadwal operasi	
Aktor	Pengguna.
Tujuan	Untuk mengakses jadwal operasi pengguna.
Kondisi Sukses	Pengguna berhasil mengakses jadwal operasi.
Kondisi Gagal	Pengguna tidak dapat mengakses jadwal operasi.
Kondisi Awal	Pengguna telah masuk kedalam aplikasi dan berada di halaman awal.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan jadwal operasi. 2. Sistem menampilkan halaman jadwal operasi pengguna.

Tabel 4.8 Skenario melihat detail operasi

Melihat detail operasi	
Aktor	Pengguna.
Tujuan	Untuk mengakses detail operasi pengguna.
Kondisi Sukses	Pengguna berhasil mengakses detail operasi yang dipilih.
Kondisi Gagal	Pengguna tidak dapat mengakses detail operasi yang dipilih.
Kondisi Awal	Pengguna telah memilih pilihan jadwal operasi pada halaman utama.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar operasi yang ada. 2. Sistem menampilkan halaman detail visite yang dipilih pengguna.

Tabel 4.9 Skenario mengonfirmasi visite

Mengonfirmasi visite	
Aktor	Pengguna.
Tujuan	Untuk mengubah status visite menjadi telah dilaksanakan.
Kondisi Sukses	Status visite berubah di <i>database</i> .
Kondisi Gagal	Status visite tidak berubah di <i>database</i> .
Kondisi Awal	Pengguna telah memilih jadwal pasien pada halaman visite.
Alir Utama	<ol style="list-style-type: none"> 1. Pengguna memilih tombol konfirmasi visite. 2. Sistem menampilkan pesan status visite sudah berubah.

Tabel 4.10 Skenario mengubah status operasi

Mengubah status operasi	
Aktor	Pengguna.
Tujuan	Untuk mengubah status operasi .

Kondisi Sukses	Status operasi berubah di <i>database</i> .
Kondisi Gagal	Status operasi tidak berubah di <i>database</i> .
Kondisi Awal	Pengguna telah memilih jadwal operasi pada halaman operasi.
Alir Utama	1. Pengguna memilih tombol status operasi selesai. 2. Sistem menampilkan pesan status visite sudah berubah.
Alir Alternatif	1. Pengguna memilih tombol status operasi ekstensi. 2. Sistem menampilkan pesan status visite sudah berubah.

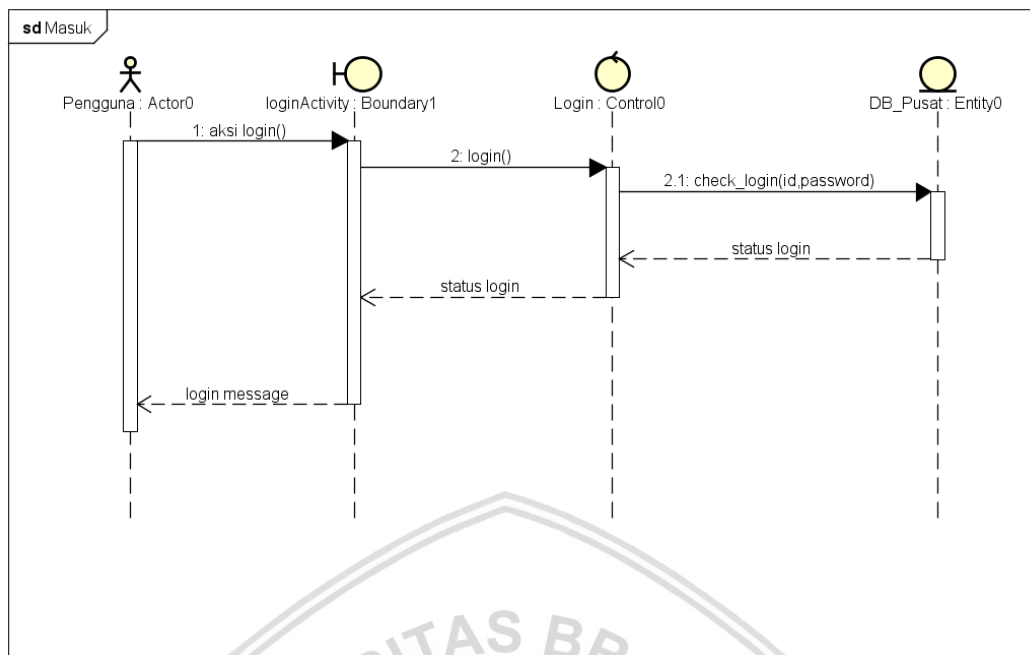
4.2 Perancangan

4.2.1 Sequence Diagram

Menurut Sommerville (2011) diagram *sequence* adalah perancangan alur proses pada aplikasi yang digambarkan berdasarkan waktu dalam bentuk diagram. Setiap diagram yang digambarkan merupakan fungsi yang terdapat didalam *use case*.

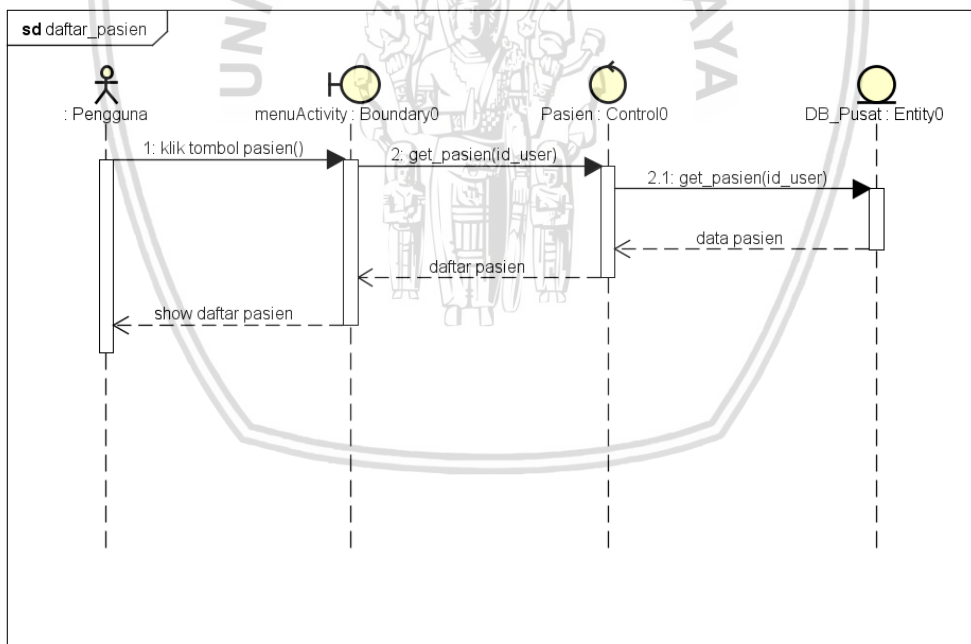
1. Diagram *sequence* masuk

Pada Gambar 4.5, fungsi masuk akan diinisiasi oleh pengguna ketika pengguna telah memasukkan *username* dan *password*, lalu menekan tombol “masuk”. Permintaan pengguna akan diproses melalui fungsi `login()` pada kelas Login. Kelas Login kemudian memanggil fungsi `check_login()` pada *database* pusat Rumah Sakit XYZ melalui *web service* untuk mendapatkan konfirmasi identitas *login* pengguna. Status *login* dari *database* pusat kemudian dikembalikan lagi ke kelas Login dan ditampilkan ke pengguna. Jika status login berhasil, maka pengguna akan diarahkan ke halaman utama aplikasi.



Gambar 4.5 Sequence diagram masuk

2. Diagram sequence daftar pasien

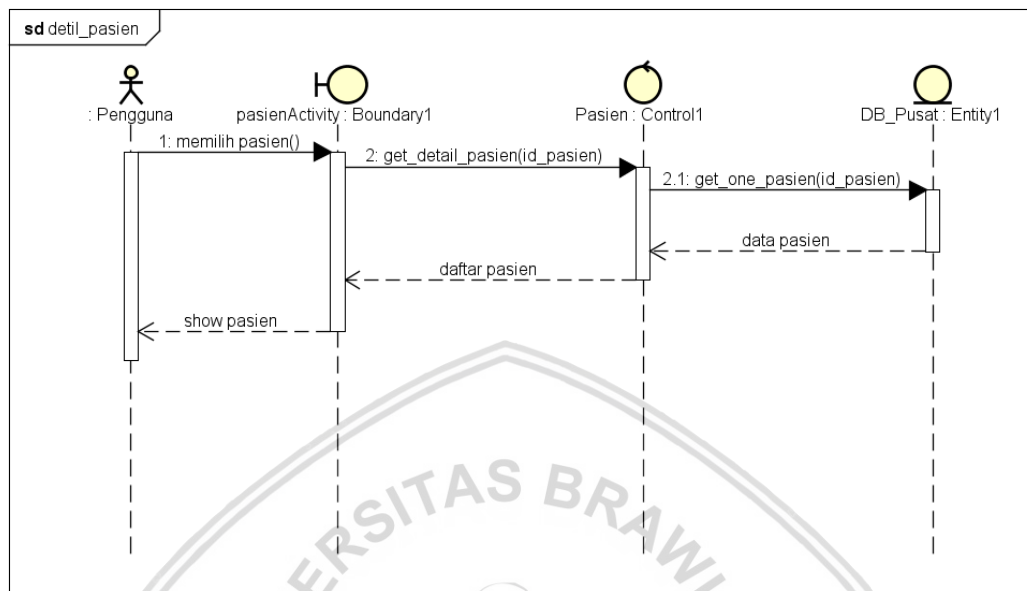


Gambar 4.6 Sequence diagram daftar pasien

Pada Gambar 4.6, fungsi daftar pasien akan diinisiasi oleh pengguna ketika pengguna memilih tombol pasien pada aplikasi. Permintaan pengguna kemudian akan diproses oleh kelas Pasien melalui fungsi `get_pasien()` dengan parameter `id_user`. Kelas Pasien kemudian memanggil fungsi `get_pasien()`

dari *database pusat*, yang kemudian akan memberikan balasan berupa data pasien untuk ditampilkan kembali kepada pengguna dalam bentuk *list*/daftar.

3. Diagram *sequence* detail pasien

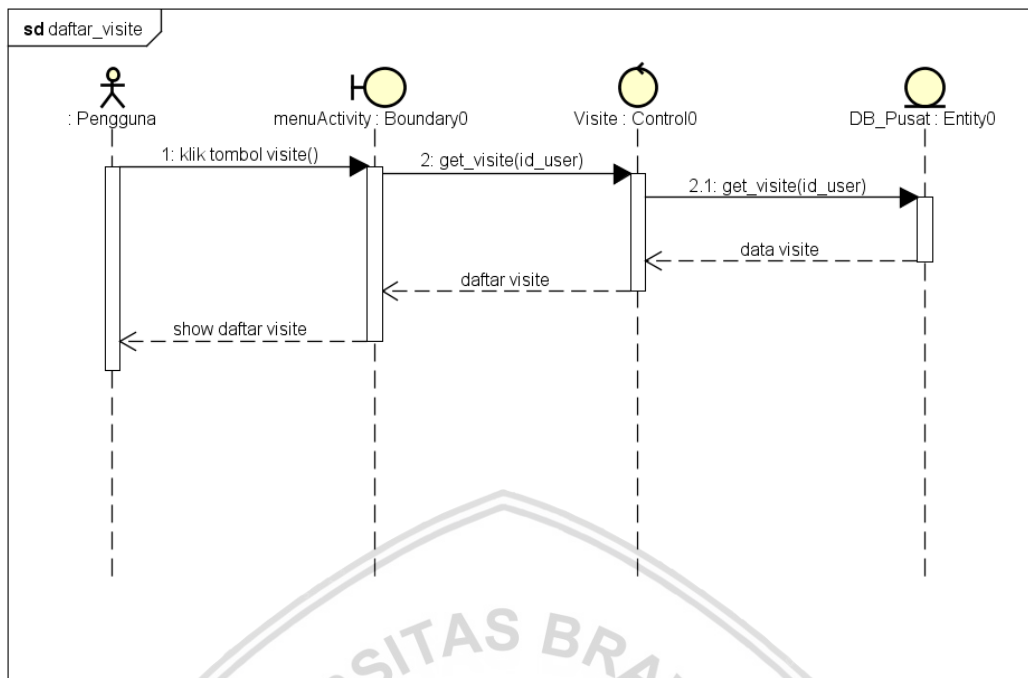


Gambar 4.7 Sequence diagram detail pasien

Pada Gambar 4.7, fungsi detail pasien akan diinisiasi oleh pengguna ketika pengguna memilih salah satu pasien dari daftar pasien yang ada. Permintaan pengguna kemudian akan diproses oleh kelas Pasien melalui fungsi `get_detail_pasien()` dengan parameter `id_pasien`. Kelas Pasien kemudian memanggil fungsi `get_one_pasien()` dari database pusat, untuk kemudian mengembalikan data detail pasien yang diminta oleh pengguna untuk ditampilkan.

4. Diagram *sequence* daftar visite

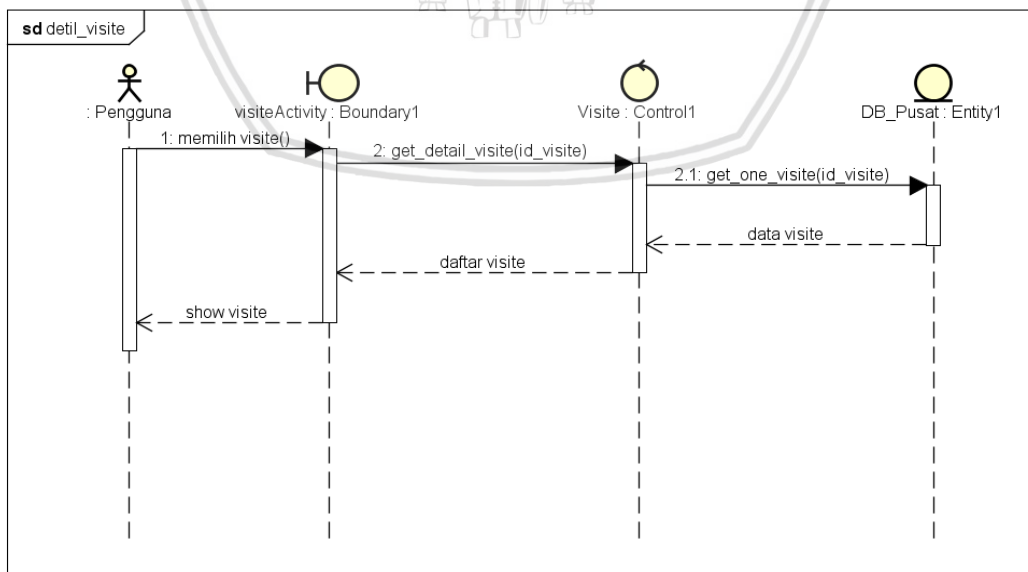
Pada Gambar 4.8, fungsi daftar visite akan diinisiasi oleh pengguna ketika pengguna memilih tombol visite pada aplikasi. Permintaan pengguna kemudian akan diproses oleh kelas Visite melalui fungsi `get_visite()` dengan parameter `id_user`. Kelas Visite kemudian memanggil fungsi `get_visite()` dari *database pusat*, yang kemudian akan memberikan balasan berupa data visite untuk ditampilkan kembali kepada pengguna dalam bentuk *list*/daftar.



Gambar 4.8 Sequence diagram daftar visite

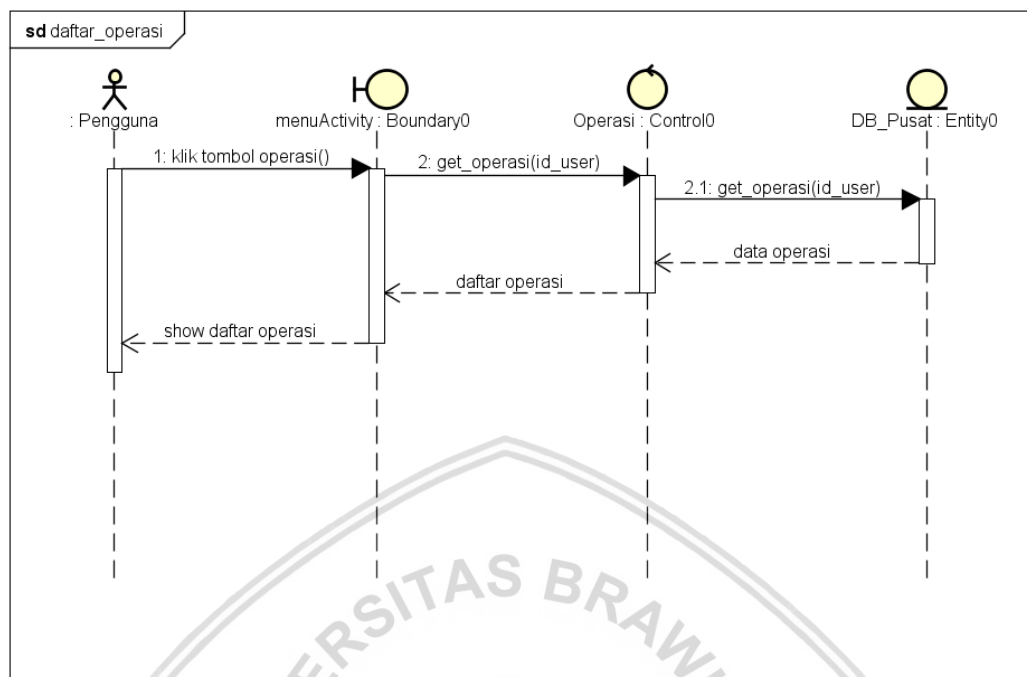
5. Diagram sequence detail visite

Pada Gambar 4.9, fungsi detail visite akan diinisiasi oleh pengguna ketika pengguna memilih salah satu pasien dari daftar visite yang ada. Permintaan pengguna kemudian akan diproses oleh kelas Visite melalui fungsi `get_detail_visite()` dengan parameter `id_visite`. Kelas Visite kemudian memanggil fungsi `get_one_visite()` dari database pusat, untuk kemudian mengembalikan data detail visite yang diminta oleh pengguna untuk ditampilkan.



Gambar 4.9 Sequence diagram detail visite

6. Diagram *sequence* daftar operasi

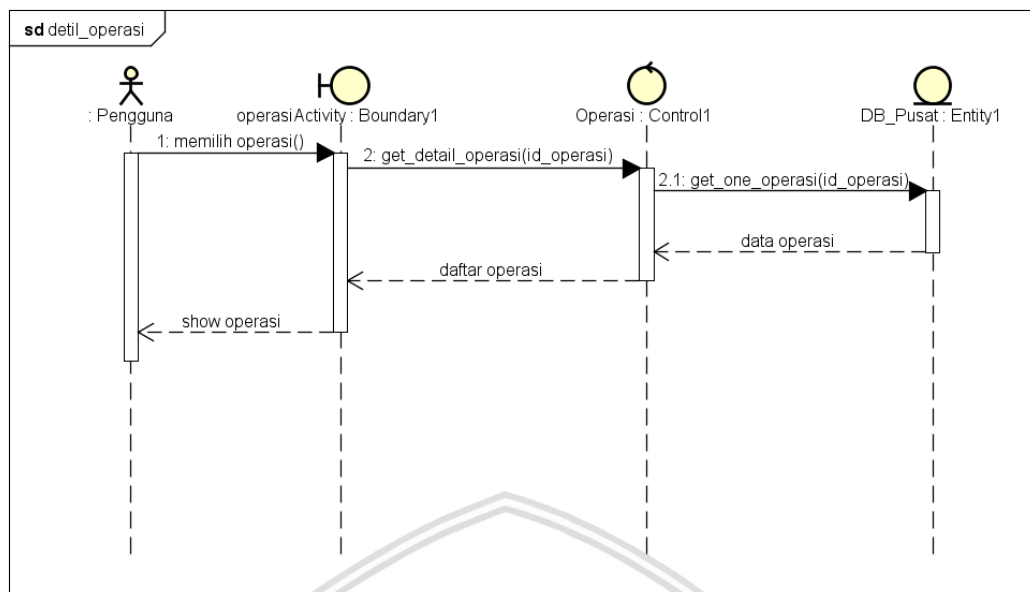


Gambar 4.10 Sequence diagram daftar operasi

Pada Gambar 4.10, fungsi daftar operasi akan diinisiasi oleh pengguna ketika pengguna memilih tombol operasi pada aplikasi. Permintaan pengguna kemudian akan diproses oleh kelas Operasi melalui fungsi `get_operasi ()` dengan parameter `id_user`. Kelas Operasi kemudian memanggil fungsi `get_operasi ()` dari *database pusat*, yang kemudian akan memberikan balasan berupa data operasi untuk ditampilkan kembali kepada pengguna dalam bentuk *list/daftar*.

7. Diagram *sequence* detail operasi

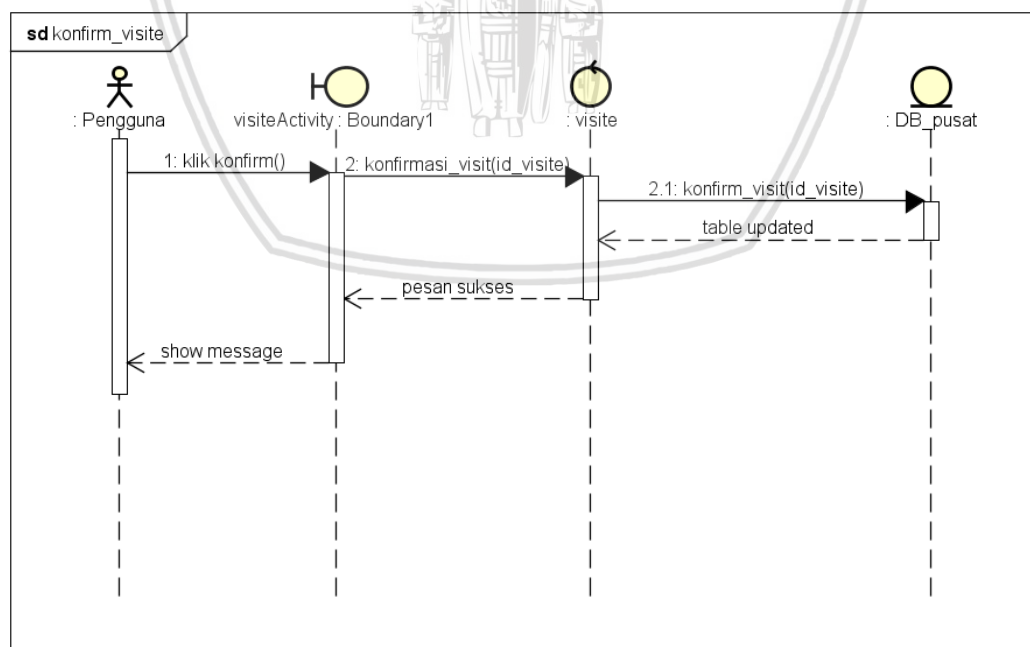
Pada Gambar 4.11, fungsi detail operasi akan diinisiasi oleh pengguna ketika pengguna memilih salah satu jadwal operasi dari daftar yang ada. Permintaan pengguna kemudian akan diproses oleh kelas Operasi melalui fungsi `get_detail_operasi ()` dengan parameter `id_visite`. Kelas Operasi kemudian memanggil fungsi `get_one_operasi()` dari database pusat, untuk kemudian mengembalikan data detail visite yang diminta oleh pengguna untuk ditampilkan.



Gambar 4.11 Sequence diagram detil operasi

8. Diagram *sequence* konfirmasi visite

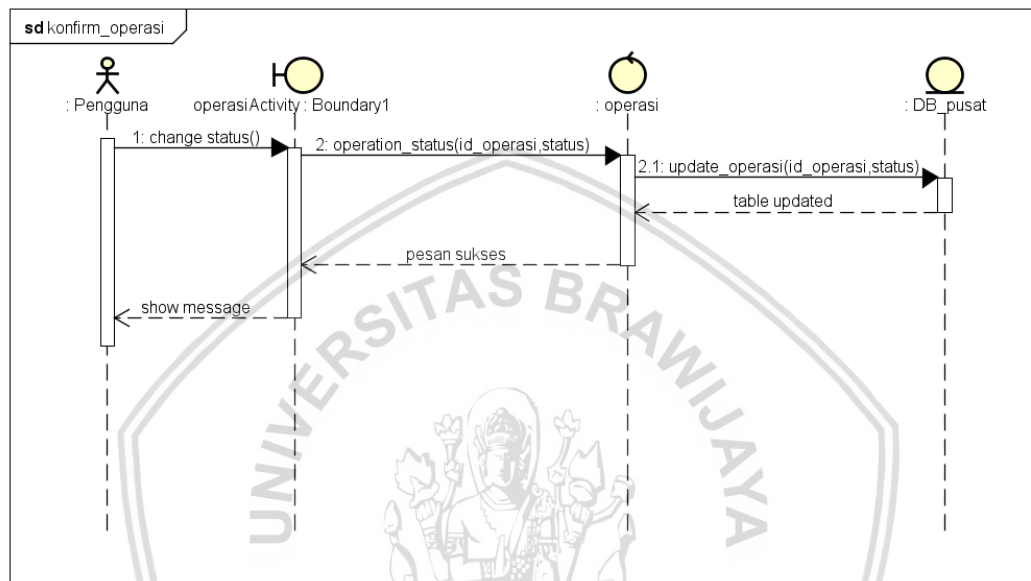
Pada Gambar 4.12, fungsi konfirmasi visite akan diinisiasi oleh pengguna ketika pengguna memilih tombol konfirmasi pada halaman visite. Fungsi `konfirmasi_visit()` akan dipanggil di kelas visite dengan parameter `id_visite`, yang kemudian akan memanggil fungsi `konfirmasi_visit()` dari Database pusat Rumah Sakit XYZ, yang akan mengubah status visite pasien di *database*. Pesan sukses kemudian ditampilkan kembali ke pengguna.



Gambar 4.12 Sequence diagram konfirmasi visite

9. Diagram *sequence* konfirmasi tindakan operasi

Pada Gambar 4.13, fungsi konfirmasi operasi akan diinisiasi oleh pengguna ketika pengguna memilih tombol konfirmasi pada halaman operasi. Fungsi `operation_status()` akan dipanggil di kelas `visite` dengan parameter `id_visite` dan `status` yang kemudian akan memanggil fungsi `update_operasi ()` dari Database pusat Rumah Sakit XYZ, yang akan mengubah status operasi pasien di *database*. Pesan sukses kemudian ditampilkan kembali ke pengguna.



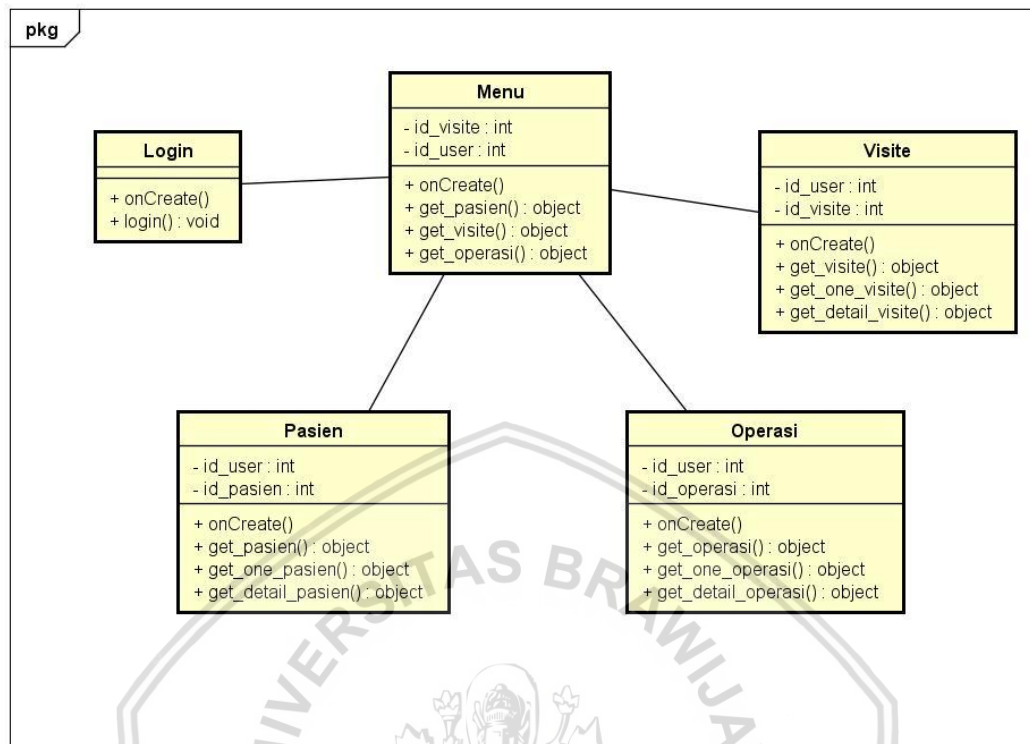
Gambar 4.13 *Sequence diagram* konfirmasi operasi

4.2.2 Class Diagram

Seperti yang tertera pada Gambar 4.14, terdapat 5 kelas utama dalam pembangunan aplikasi.

1. Login: kelas yang berisi fungsi untuk mengakses operasi login.
2. Menu: kelas yang menjembatani antara pengguna dengan fungsi-fungsi utama aplikasi, yaitu fungsi pasien, visite, dan operasi.
3. Visite: kelas untuk mengakses fungsi pada halaman visite dan detil visite.
4. Pasien: kelas untuk mengakses fungsi pada halaman pasien dan detil pasien.

5. Operasi: kelas untuk mengakses fungsi pada halaman operasi.



Gambar 4.14 class diagram aplikasi

4.2.3 Perancangan Antar Muka

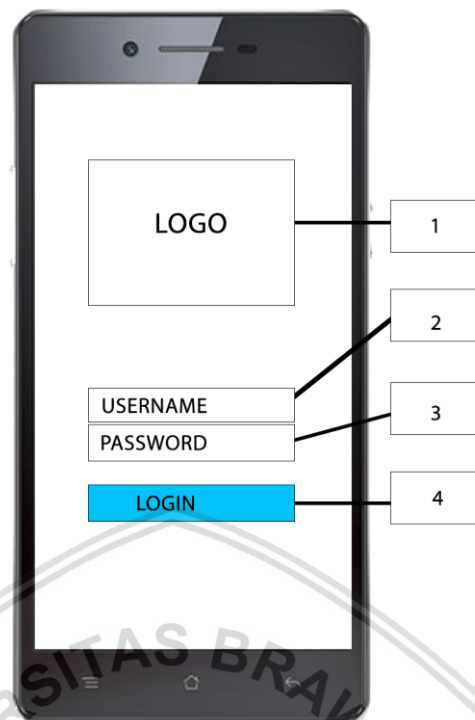
Pada bagian ini akan dijelaskan rancangan antar muka aplikasi yang akan dibangun. Dalam rancangan, komponen aplikasi dipisahkan dan diberi nomor untuk dijelaskan pada bagian keterangan.

1. Halaman Login

Rancangan halaman login pada Gambar 4.15 adalah halaman diakses ketika pertama kali menjalankan aplikasi. Halaman ini terdiri dari 3 komponen utama, yaitu gambar untuk logo, kolom input untuk username & password, dan tombol untuk login.

Keterangan:

1. Logo: yaitu gambar yang menampilkan logo Rumah Sakit XYZ Malang.
2. Kolom username: Kolom untuk memasukkan username pengguna yang telah terdaftar.
3. Kolom password: Kolom untuk memasukkan password pengguna yang telah terdaftar.
4. Tombol login: Tombol untuk melakukan permintaan masuk kedalam aplikasi berdasarkan username dan password yang telah diinputkan.



Gambar 4.15 Rancangan antar muka login

2. Halaman utama

Rancangan halaman utama pada Gambar 4.16 adalah halaman yang diakses ketika pengguna telah berhasil melakukan proses login. Halaman ini terdiri dari 1 komponen utama yaitu tombol untuk mengakses halaman pasien, visite, dan operasi.

Keterangan:

1. Tombol pasien: Tombol yang digunakan untuk mengakses halaman pasien.
2. Tombol visite: Tombol yang digunakan untuk mengakses halaman visite.
3. Tombol operasi: Tombol yang digunakan untuk mengakses halaman operasi.

3. Halaman daftar pasien

Rancangan halaman daftar pasien pada Gambar 4.17 adalah halaman yang diakses ketika pengguna memilih tombol pasien di halaman utama. Halaman ini terdiri dari 1 komponen utama yaitu tombol untuk mengakses halaman detail pasien yang dipilih.

Keterangan:

1. Keterangan: yaitu tulisan penunjuk bahwa pengguna berada di halaman daftar pasien.
2. Tombol pasien: yaitu daftar nama-nama pasien yang diurutkan dalam urutan tertentu, yang dapat diklik untuk mengakses detail halaman pasien yang dipilih.



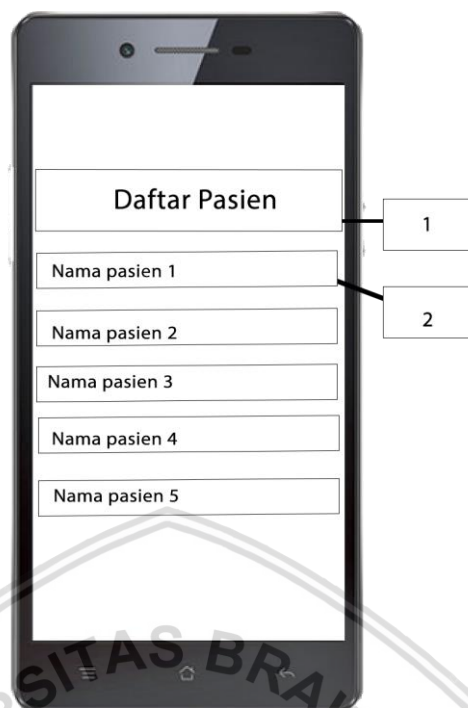
Gambar 4.16 Rancangan antar muka halaman utama

4. Halaman detil pasien

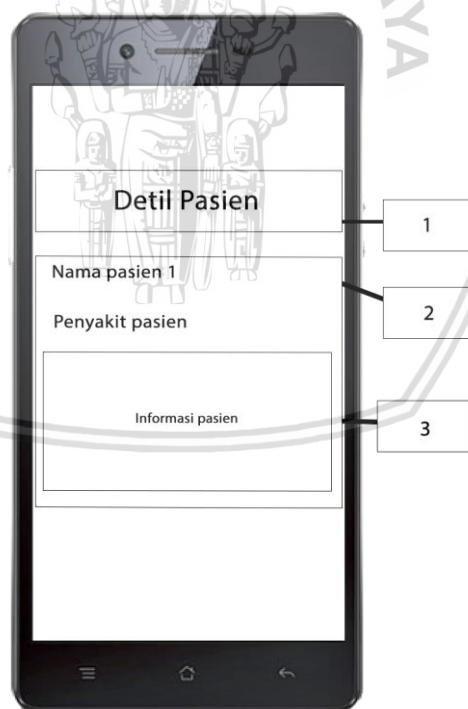
Rancangan halaman daftar pasien pada Gambar 4.18 adalah halaman yang diakses ketika pengguna memilih salah satu pasien pada halaman daftar pasien. Halaman ini berisi informasi tentang pasien yang dipilih. Informasi yang ditampilkan adalah informasi yang didapatkan dari database pusat Rumah Sakit XYZ. Tidak ada interaksi khusus yang dapat dilakukan pengguna pada halaman ini.

Keterangan:

1. Nama halaman: yaitu tulisan penunjuk bahwa pengguna berada di halaman detil pasien.
2. Informasi umum pasien: yaitu bagian yang menampilkan informasi umum pasien seperti nama dan penyakit yang diidap.
3. Informasi khusus pasien: yaitu bagian yang menampilkan informasi khusus terkait pasien, seperti riwayat obat, durasi penyakit yang diidap, dan lain sebagainya.



Gambar 4.17 Rancangan antar muka halaman daftar pasien



Gambar 4.18 Rancangan antar muka halaman detil pasien

5. Halaman daftar visite

Rancangan halaman daftar pasien pada Gambar 4.19 adalah halaman yang diakses ketika pengguna memilih tombol visite di halaman utama. Halaman ini

terdiri dari 1 komponen utama yaitu tombol untuk mengakses halaman visite yang dipilih.

Keterangan:

1. Keterangan: yaitu tulisan penunjuk bahwa pengguna berada di halaman daftar visite.
2. Tombol pasien visite: yaitu daftar nama-nama pasien yang diurutkan dalam urutan tertentu, yang dapat diklik untuk mengakses detail halaman visite pasien yang dipilih. Pada tombol juga terdapat waktu yang dijadwalkan bagi pengguna untuk melakukan visite.



Gambar 4.19 Rancangan antar muka halaman daftar visite

6. Halaman detail visite

Rancangan halaman detail visite pada Gambar 4.20 adalah halaman yang diakses ketika pengguna memilih salah satu pasien pada halaman daftar visite. Halaman ini berisi informasi tentang visite yang dipilih. Informasi yang ditampilkan adalah informasi yang didapatkan dari database pusat Rumah Sakit XYZ. Tidak ada interaksi khusus yang dapat dilakukan pengguna pada halaman ini.

Keterangan:

1. Nama halaman: yaitu tulisan penunjuk bahwa pengguna berada di halaman detail visite.
2. Informasi umum pasien: yaitu bagian yang menampilkan informasi umum pasien seperti nama pasien, ruang rawat inap pasien, dan waktu visite pasien.

3. Informasi khusus pasien: yaitu bagian yang menampilkan informasi khusus terkait pasien, seperti riwayat obat, durasi rawat inap, penyakit yang diidap, dan lain sebagainya.

4. Tombol konfirmasi: yaitu tombol untuk mengonfirmasi bahwa visite telah dilakukan ke pasien tertentu.



Gambar 4.20 Rancangan antar muka halaman detil visite

7. Halaman daftar operasi

Rancangan halaman daftar operasi pada Gambar 4.21 adalah halaman yang diakses ketika pengguna memilih tombol operasi di halaman utama. Halaman ini terdiri dari 1 komponen utama yaitu tombol untuk mengakses halaman detil operasi yang dipilih.

Keterangan:

1. Keterangan: yaitu tulisan penunjuk bahwa pengguna berada di halaman daftar operasi.

2. Tombol operasi: yaitu daftar nama-nama pasien operasi yang diurutkan dalam urutan tertentu, yang dapat diklik untuk mengakses detil halaman operasi pasien yang dipilih. Pada tombol juga terdapat waktu yang dijadwalkan bagi pengguna untuk melakukan operasi dan jenis operasi yang akan dilakukan.



Gambar 4.21 Rancangan antar muka halaman daftar operasi

8. Halaman detail operasi

Rancangan halaman detail pasien pada Gambar 4.22 adalah halaman yang diakses ketika pengguna memilih salah satu pasien/operasi pada halaman daftar operasi. Halaman ini berisi informasi tentang operasi yang dipilih. Informasi yang ditampilkan adalah informasi yang didapatkan dari database pusat Rumah Sakit XYZ. Tidak ada interaksi khusus yang dapat dilakukan pengguna pada halaman ini.

Keterangan:

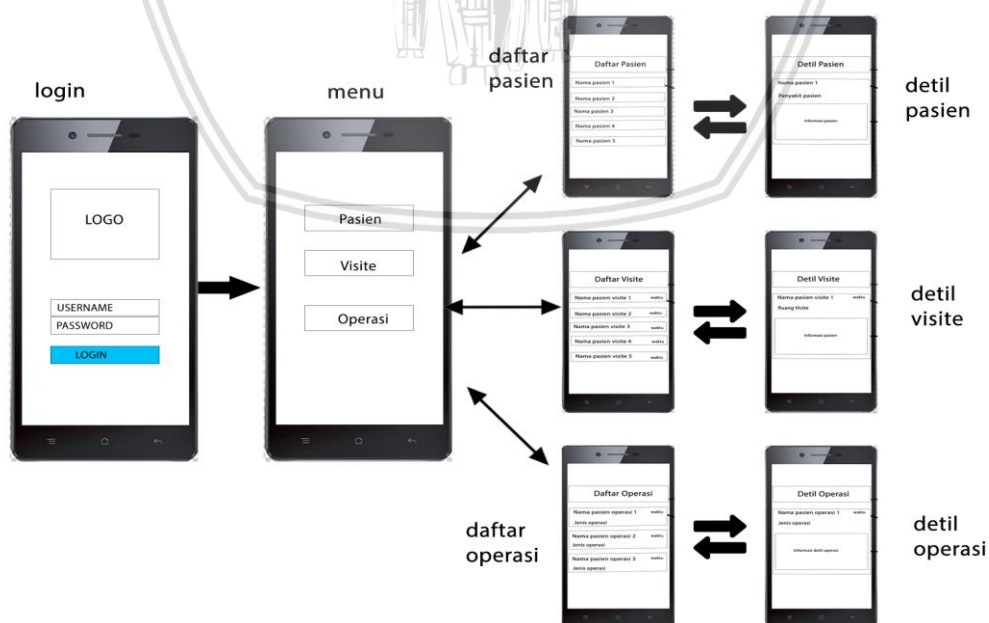
1. Nama halaman: yaitu tulisan penunjuk bahwa pengguna berada di halaman detail operasi.
2. Informasi umum operasi: yaitu bagian yang menampilkan informasi umum pasien seperti nama pasien, jenis operasi, ruang operasi, dan waktu operasi.
3. Informasi khusus pasien: yaitu bagian yang menampilkan informasi khusus terkait operasi, tindakan yang akan dilakukan, perkiraan durasi operasi, dan lain sebagainya.
4. Tombol status: yaitu tombol untuk mengonfirmasi atau melakukan ekstensi tindakan operasi yang dilakukan.



Gambar 4.22 Rancangan antar muka halaman detil operasi

4.2.4 Perancangan *Screenflow*

Screenflow/alur layar pada aplikasi yang dibangun tertera pada Gambar 4.23. Pada screenflow tergambar aliran halaman yang berpindah-pindah tergantung dari interaksi user pada halaman yang tampil.

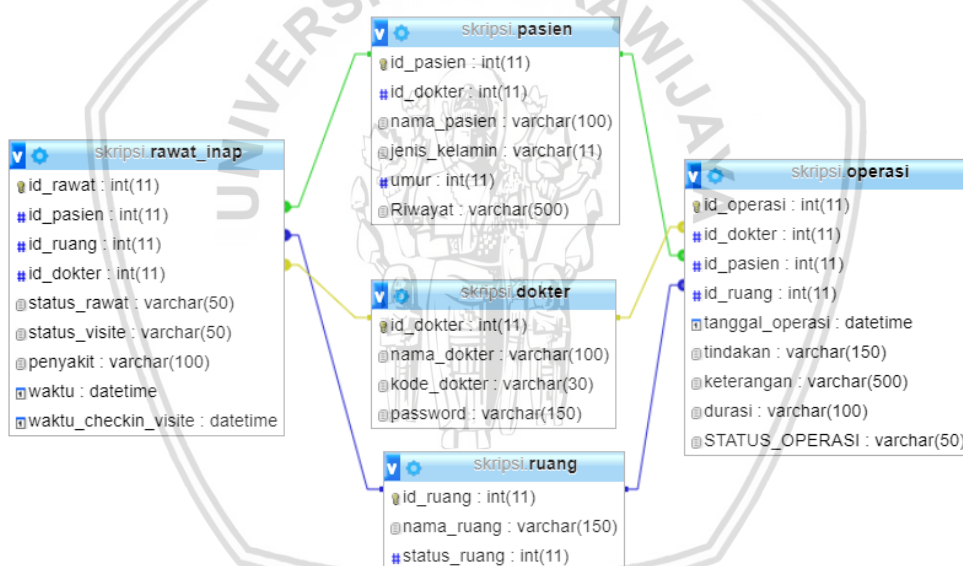


Gambar 4.23 *Screenflow* aplikasi

4.2.5 Perancangan Data

Data terkait perancangan struktur data didapatkan dari proses diskusi dan wawancara dengan departemen IT Rumah Sakit XYZ. Struktur data yang dirancang bukan merupakan struktur data sebenarnya yang ada di RS XYZ. Atribut tiap entitas didapatkan dari struktur basis data di Rumah Sakit XYZ yang memiliki relevansi terhadap permasalahan penelitian. Struktur data ini digunakan untuk pengujian aplikasi secara lokal dan sebagai acuan pengujian aplikasi.

Perancangan struktur data pada aplikasi ini seperti yang tergambar pada Gambar 4.24. Terdapat 3 objek utama dalam struktur data yang dibangun, yaitu pasien, dokter, dan ruang. Objek pasien memuat informasi yang berhubungan terkait pasien seperti nama, dan umur. Objek dokter memuat informasi tenaga medis dokter, seperti nama, dan identitas untuk masuk kedalam aplikasi. Objek ruang memuat informasi kamar rawat inap pasien dan ruang tindakan operasi. Objek operasi memuat data tindakan operasi yang dilakukan seperti dokter yang melakukan tindakan, pasien yang ditindak, dan ruangan operasi. Objek rawat inap memuat informasi ruangan, pasien yang dirawat, dan penyakit yang diidap.



Gambar 4.24 Struktur data

BAB 5 IMPLEMENTASI

5.1 Implementasi

Pada bagian ini akan dijelaskan mengenai hal-hal yang dilakukan dan dipersiapkan dalam proses implementasi sistem berdasarkan hasil perancangan. Persiapan implementasi meliputi penentuan lingkungan pengembangan aplikasi berupa perangkat keras maupun perangkat lunak, dan batasan implementasi. Pelaksanaan implementasi meliputi implementasi basis data, *class*, antar muka, dan kodeSpesifikasi Perangkat Keras.

5.2 Spesifikasi Perangkat Lunak

Dalam implementasi aplikasi, spesifikasi perangkat lunak untuk proses pengembangan dan proses pengujian seperti yang tercantum pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat lunak

Perangkat pengembangan	Perangkat pengujian
Sistem Operasi: Windows 8.1 <i>SDK</i> : Android Studio 2.2.3 <i>Web service</i> : MySQL 5.6.24 , PHP 5.6.8	Sistem Operasi: Android 4.4.2

5.3 Spesifikasi Perangkat Keras

Dalam implementasi aplikasi, spesifikasi perangkat keras untuk proses pengembangan dan proses pengujian seperti yang tercantum pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat keras

Perangkat pengembangan	Perangkat pengujian
Prosesor: AMD A10 RAM: 12 GB VGA: ATI Radeon 8250M	Prosesor: Quad-core 1.3 GHz Cortex-A7 Ram: 1GB

5.4 Batasan Implementasi

Dalam implementasi, batasan yang diterapkan terletak pada proses pengambilan data. Data pasien dan data dokter yang digunakan dalam pengimplementasian aplikasi merupakan data *dummy* yang diakses melalui JSON yang diunggah secara publik ke web melalui www.myjson.com. Dalam implementasi tidak ada akses langsung ke basis data RS XYZ untuk memanipulasi

data, pengimplementasian struktur data merupakan bentuk dasar dari struktur data sebenarnya yang digunakan dalam infrastruktur data RS XYZ.

Dalam implementasi ini versi Android yang dijadikan acuan adalah Android KitKat versi 4.4.2 keatas, karena fungsionalitas versi Android ini dapat digunakan oleh seluruh versi Android diatasnya yang mencakup mayoritas dari seluruh perangkat Android yang ada di pasaran. Penggunaan Android versi 4 juga disebabkan oleh belum ada data yang menjabarkan penggunaan versi *smartphone* oleh tenaga medis di RS XYZ, sehingga dipilih Android versi 4 yang merupakan versi aman untuk implementasi aplikasi agar dapat digunakan di mayoritas *smartphone* dengan sistem operasi Android.

5.5 Implementasi Basis Data

Berikut merupakan penjabaran dari tiap tabel yang digunakan untuk implementasi aplikasi. Implementasi entitas dan atribut tabel disesuaikan dengan preancangan struktur data pada Gambar 4.24. Tabel-tabel yang dijabarkan merepresentasikan struktur dasar data yang digunakan di RS XYZ, dan bukanlah struktur data sebenarnya.

5.5.1 Dokter

Tabel 5.3 Tabel dokter

Nama Kolom	Tipe Data
id_dokter(primary key)	int
nama_dokter	Varchar(100)
kode_dokter	Varchar(30)
password	Varchar(150)

Tabel 5.3 adalah tabel yang digunakan untuk menyimpan data dokter yang akan menggunakan aplikasi. Tabel ini menyimpan nomor ID dokter dan password yang digunakan untuk keperluan *login* pada aplikasi.

5.5.2 Pasien

Tabel 5.4 Tabel pasien

Nama Kolom	Tipe Data
id_pasien(primary key)	int
id_dokter(foreign key)	int
nama_pasien	Varchar(100)
jenis_kelamin	Varchar(11)
umur	Int

riwayat	Varchar(500)
---------	--------------

Tabel 5.4 menyimpan data dasar pasien seperti nama, umur, jenis kelamin, dan riwayat penyakit yang diidap. Tiap pasien memiliki seorang dokter utama yang dipasangkan untuk menangani pasien dalam proses visite.

5.5.3 Ruang

Tabel 5.5 Tabel ruang

Nama Kolom	Tipe Data
id_ruang(primary key)	int
nama_ruang	Varchar(150)
status_ruang	int

Tabel 5.5 memuat data dasar ruangan yang ada di rumah sakit. Ruangan yang tercantum dalam tabel ini adalah ruang rawat inap, dan ruang operasi. Kolom status_ruang digunakan untuk mendapatkan ketersediaan ruangan yang dimaksud.

5.5.4 Rawat Inap

Tabel 5.6 Tabel rawat inap

Nama Kolom	Tipe Data
id_rawat(primary key)	int
id_pasien(foreign key)	int
id_dokter(foreign key)	int
id_ruang(foreign key)	int
status_rawat	Varchar(50)
status_visite	Varchar(50)
penyakit	Varchar(100)
waktu	datetime
waktu_checkin_visite	datetime

Untuk setiap pasien yang melakukan proses rawat inap, data pasien tersebut akan tersimpan di tabel rawat_inap seperti yang tercantum pada Tabel 5.6. Dalam tabel ini terdapat data penyakit yang diidap pasien, data status rawat yaitu apakah pasien masih dirawat atau tidak, status visite yaitu apakah visite

dokter pada hari tertentu telah terlaksana atau belum, dan waktu. Kolom waktu disini berfungsi untuk mencatat waktu visite yang dianjurkan untuk diikuti oleh dokter yang dipasangkan kepada seorang pasien. kolom waktu *check-in* merupakan waktu yang dicatat ketika dokter melakukan visite pada seorang pasien.

5.5.5 Operasi

Tabel 5.7 Tabel operasi

Nama Kolom	Tipe Data
id_operasi(primary key)	int
id_pasien(foreign key)	int
id_dokter(foreign key)	int
id_ruang(foreign key)	int
tanggal_operasi	datetime
tindakan	Varchar(150)
keterangan	Varchar(500)
durasi	Varchar(100)
status_operasi	Varchar(50)

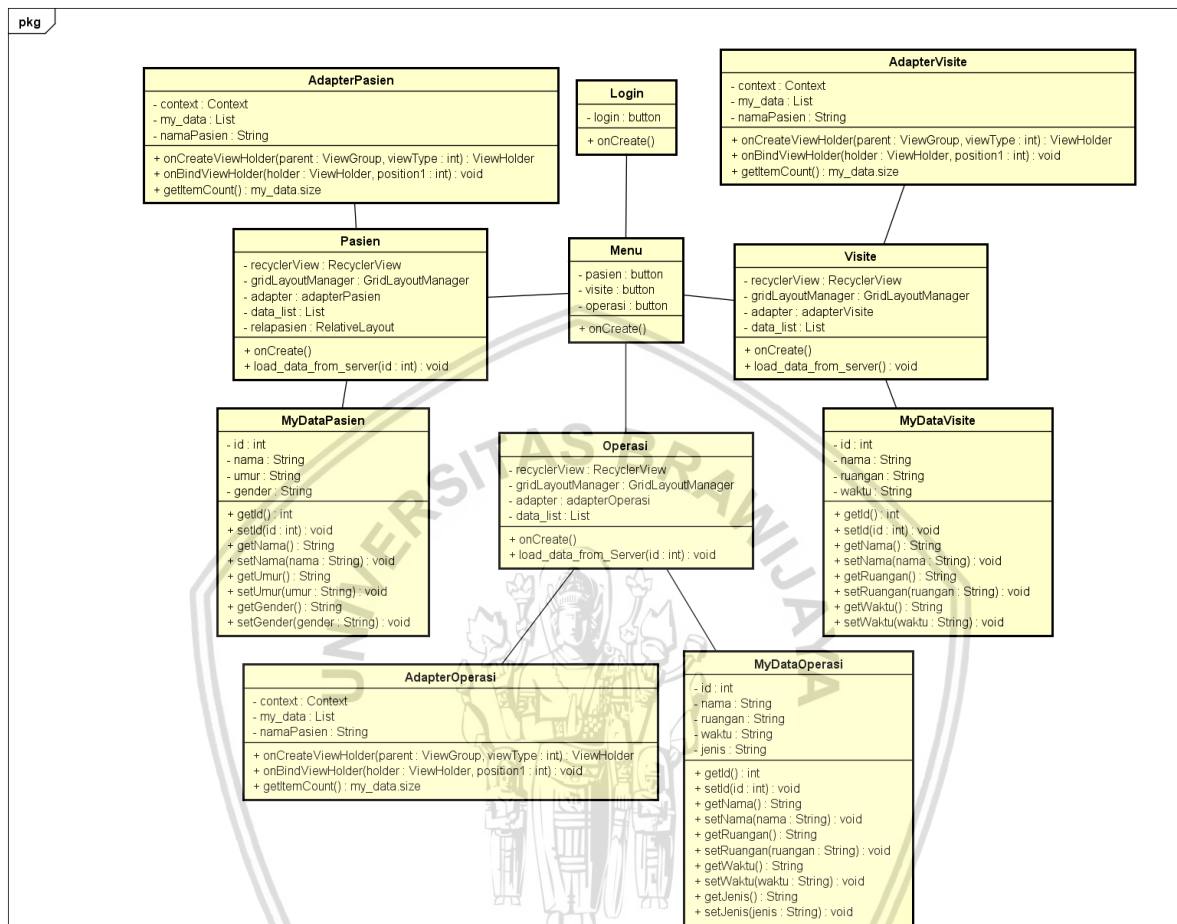
Tabel 5.7 memuat informasi mengenai tindakan operasi yang dilakukan untuk pasien. Kolom tindakan memuat informasi mengenai jenis operasi yang dilakukan. Kolom keterangan memuat informasi mengenai catatan operasi dan atau tim dokter yang mengikuti operasi.

5.6 Implementasi Class

Dalam implementasi kelas terdapat 3 kelas utama yaitu kelas Pasien, Operasi, dan Visite. Masing-masing dari kelas utama ini membuat objek *RecyclerView* yang digunakan sebagai *container* untuk data yang ditampilkan di aplikasi. Data yang ditampilkan pada *RecyclerView* merupakan data yang tersimpan pada *List* dari masing-masing kelas *MyData*. Pada kelas *MyData* terdapat fungsi *setter* dan *getter* untuk masing masing variabel yang digunakan. Kelas *MyData* dihubungkan dengan menggunakan kelas *Adapter* untuk tiap kelas utama. Kelas *Adapter* juga berfungsi untuk menghubungkan *List* dengan *layout* dari tiap fungsi. Di dalam kelas *MyData* terdapat variabel-variabel dasar informasi objek tertentu, seperti *id*, nama, dan lain lain.

Pada kelas utama memanggil fungsi *load_from_server* yang merupakan fungsi untuk memanggil webservice untuk mendapatkan data untuk ditampilkan dari server ke aplikasi melalui manipulasi data JSON. Kelas *Menu* digunakan untuk

menjembatani antara tombol-tombol pada tampilan menu dengan halaman pada kelas utama. Kelas *Login* menjembatani antara halaman awal aplikasi dengan verifikasi basis data untuk masuk ke dalam aplikasi.



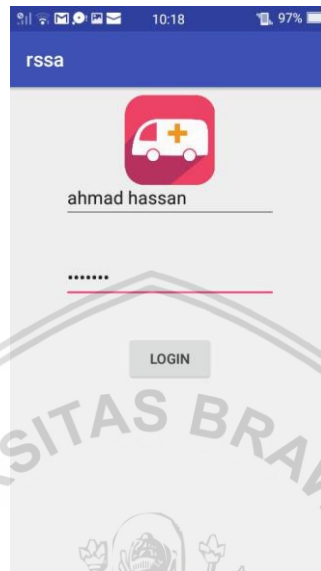
Gambar 5.1 Diagram implementasi kelas

5.7 Implementasi Antar Muka

Pada bagian ini merupakan hasil implementasi antar muka untuk aplikasi yang dikembangkan. Setiap data yang tampil pada halaman bukanlah representasi data sebenarnya, melainkan hanya data contoh yang menggunakan basis data *dummy*. Potongan gambar implementasi antar muka diambil dari perangkat dengan resolusi 480 X 854 dpi dengan ukuran layar 4,5 inci .

5.7.1 Halaman Login

Implementasi halaman login pada Gambar 5.2 mengacu pada *mockup* perancangan pada Gambar 4.15. Pada *layout* digunakan 2 buah objek *EditText* untuk kolom *username* dan *password*, dan satu buah objek *Button* untuk mengirimkan identitas *login* dan mengakses halaman menu.



Gambar 5.2 Antar muka halaman Login

5.7.2 Halaman Menu

Implementasi halaman menu pada Gambar 5.3 mengacu pada *mockup* perancangan pada Gambar 4.16. Pada *layout* terdapat 3 buah objek *Button* yang masing-masing akan mengarahkan ke halaman tertentu pada aplikasi.



Gambar 5.3 Antar muka halaman Menu

5.7.3 Halaman Daftar Pasien

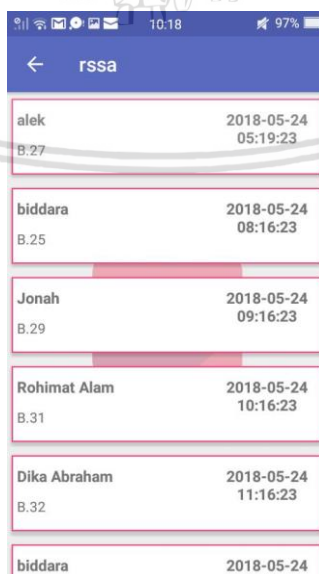
Implementasi halaman daftar pasien pada Gambar 5.4 mengacu pada *mockup* perancangan pada Gambar 4.17. Pada *layout* digunakan objek *RecyclerView* untuk menampilkan daftar pasien yang dapat di *scroll* dan secara dinamis menampilkan data dari basis data.



Gambar 5.4 Antar muka halaman daftar pasien

5.7.4 Halaman Daftar Visite

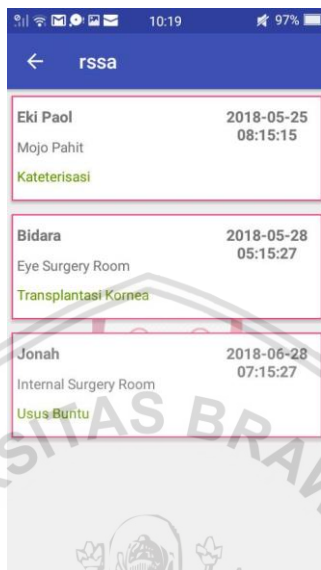
Implementasi halaman daftar pasien pada Gambar 5.5 mengacu pada *mockup* perancangan pada Gambar 4.19. Pada *layout* digunakan objek *RecyclerView* untuk menampilkan daftar visite yang dapat di *scroll* dan secara dinamis menampilkan data dari basis data.



Gambar 5.5 Antar muka halaman daftar visite

5.7.5 Halaman Daftar Operasi

Implementasi halaman daftar pasien pada Gambar 5.6 mengacu pada *mockup* perancangan pada Gambar 4.21. Pada *layout* digunakan objek *RecyclerView* untuk menampilkan daftar operasi yang dapat di *scroll* dan secara dinamis menampilkan data dari basis data.



Gambar 5.6 Antar muka halaman daftar operasi

5.7.6 Halaman Detil Pasien

Implementasi halaman detail pasien pada Gambar 5.7 mengacu pada *mockup* perancangan pada Gambar 4.18. Pada *layout* terdapat 4 buah objek *TextView* dan 1 buah objek *TextArea* untuk menampilkan data detail terkait pasien.



Gambar 5.7 Antar muka halaman detail pasien

5.7.7 Halaman Detil Visite

Implementasi halaman detil pasien pada Gambar 5.8 mengacu pada *mockup* perancangan pada Gambar 4.20. Pada *layout* terdapat 4 buah objek *TextView* dan 1 buah objek *TextArea* untuk menampilkan data detil terkait visite. 1 buah Objek *Button* digunakan untuk mengakes fungsi konfirmasi visite.



Gambar 5.8 Antar muka halaman detil visite

5.7.8 Halaman Detil Operasi

Implementasi halaman detil pasien pada Gambar 5.9 mengacu pada *mockup* perancangan pada Gambar 4.22. Pada *layout* terdapat 7 buah objek *TextView* dan 1 buah objek *TextArea* untuk menampilkan data detil terkait operasi. 1 buah Objek *Button* digunakan untuk mengakes fungsi status operasi.



Gambar 5.9 Antar muka halaman detil operasi

5.7.9 Halaman Detil Operasi 2

Implementasi halaman detil pasien pada Gambar 5.10 merupakan lanjutan dari gambar 5.9 yang menunjukkan terdapat 1 buah objek *Button* pada bagian bawah halaman detil operasi, yang berfungsi untuk mengakses fungsi status operasi ke basis data pusat.



Gambar 5.10 Antar muka halaman detil operasi 2

5.8 Implementasi Kode

Pada bagian ini akan dijabarkan beberapa fungsi utama pada aplikasi dalam bentuk *pseudocode*. Hal ini dilakukan untuk memperpendek informasi yang dimuat dan melindungi karya intelektual dari kode yang digunakan. Dalam implementasi kode, library yang digunakan adalah *okhttp3* untuk melakukan koneksi ke *web service*, *cardview* dan *recyclerview* untuk tampilan data daftar pada aplikasi.

5.8.1 Fungsi Login

Fungsi login merupakan fungsi dasar agar pengguna dapat masuk untuk menggunakan fitur-fitur aplikasi. Dalam pengimplementasian fungsi login user perlu memasukkan *username* dan *password*, yang kemudian akan dikirimkan ke *web service* untuk mendapatkan balasan apakah identitas *login* valid atau tidak. Jika status valid maka pengguna akan dipindahkan ke halaman menu untuk mengakses fitur aplikasi lainnya.

Pseudocode: Login
<p>Mulai</p> <p>Membaca input username dan password</p> <p>Mengirim username dan password kepada web service melalui library</p> <p>Mendapatkan hasil web service berupa status login</p> <p>Jika status login OK maka melanjutkan ke halaman menu</p> <p>Jika status login gagal maka tampilkan informasi login gagal</p>

selesai

5.8.2 Fungsi Daftar Pasien

Untuk melihat daftar pasien, nomor identitas pengguna akan dicatat oleh aplikasi pada saat pengguna melakukan *login*. Nomor identitas ini kemudian akan dijadikan parameter untuk mendapatkan daftar pasien dari pengguna yang bersangkutan, melalui *web service*.

Pseudocode: Daftar Pasien
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Melakukan koneksi ke web service daftar pasien dengan parameter id pengguna</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data id pasien, nama pasien, umur pasien, dan jenis kelamin pasien dari JSON</p> <p>Menambahkan data pasien kedalam objek List dari kelas MyDataPasien</p> <p>selesai</p>

5.8.3 Fungsi Daftar Visite

Untuk melihat daftar visite, nomor identitas pengguna akan dicatat oleh aplikasi pada saat pengguna melakukan *login*. Nomor identitas ini kemudian akan dijadikan parameter untuk mendapatkan daftar visite dari pengguna yang bersangkutan, melalui *web service*.

Pseudocode: Daftar Visite
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Melakukan koneksi ke web service daftar visite dengan parameter id pengguna</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data id rawat, nama pasien, nama ruang, dan waktu dari JSON</p> <p>Menambahkan data kedalam objek List dari kelas MyDataVisite</p> <p>selesai</p>

5.8.4 Fungsi Daftar Operasi

Untuk melihat daftar operasi, nomor identitas pengguna akan dicatat oleh aplikasi pada saat pengguna melakukan *login*. Nomor identitas ini kemudian akan dijadikan parameter untuk mendapatkan daftar operasi dari pengguna yang bersangkutan, melalui *web service*.

Pseudocode: Daftar Operasi
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Melakukan koneksi ke web service daftar operasi dengan parameter id pengguna</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data id operasi, nama pasien, nama ruang, tanggal operasi, dan tindakan operasi dari JSON</p> <p>Menambahkan data operasi kedalam objek List dari kelas MyDataOperasi</p> <p>selesai</p>

5.8.5 Fungsi Detil Pasien

Untuk menampilkan informasi detil pasien identitas pengguna dan nomor identitas dari pasien yang dipilih akan disimpan dan dijadikan parameter untuk mengakses *web service*. Balasan berupa JSON dari *web service* kemudian diproses dan ditampilkan kedalam aplikasi.

Pseudocode: Detil Pasien
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Menyimpan ID dari pasien yang dipilih</p> <p>Melakukan koneksi ke web service daftar pasien dengan parameter ID pengguna dan ID pasien</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data pasien dari JSON</p> <p>Menampilkan data pada layout aplikasi</p> <p>selesai</p>

5.8.6 Fungsi Detil Visite

Untuk menampilkan informasi detil visite identitas pengguna dan nomor *ID* dari visite yang dipilih akan disimpan dan dijadikan parameter untuk mengakses *web service*. Balasan berupa JSON dari *web service* kemudian diproses dan ditampilkan kedalam aplikasi.

Pseudocode: Detil Visite
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Menyimpan ID dari visite yang dipilih</p> <p>Melakukan koneksi ke web service daftar pasien dengan parameter ID pengguna dan ID visite</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data visite dari JSON</p> <p>Menampilkan data pada layout aplikasi</p>

selesai

5.8.7 Fungsi Detil Visite

Untuk menampilkan informasi detil operasi identitas pengguna dan nomor *ID* dari visite yang dipilih akan disimpan dan dijadikan parameter untuk mengakses *web service*. Balasan berupa JSON dari *web service* kemudian diproses dan ditampilkan kedalam aplikasi.

Pseudocode: Detil Operasi
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Menyimpan ID dari operasi yang dipilih</p> <p>Melakukan koneksi ke web service daftar pasien dengan parameter ID pengguna dan ID operasi</p> <p>Mendapatkan balasan berupa JSON dari webservice</p> <p>Mengambil seluruh data operasi dari JSON</p> <p>Menampilkan data pada layout aplikasi</p> <p>selesai</p>

5.8.8 Fungsi Konfirmasi Visite

Proses konfirmasi visite dilakukan dengan memberikan *command* kepada basis data melalui *web service* untuk melakukan pembaharuan kepada status visite dari seorang pengguna, untuk visite tertentu. Hal ini dicapai dengan mendapatkan nomor identitas pengguna dan nomor *ID* visite dari proses pengambilan informasi detil visite, lalu kemudian menjadikan nomor pengguna dan nomor visite sebagai parameter untuk mengakses *web service*.

Pseudocode: Konfirmasi Visite
<p>Mulai</p> <p>Menyimpan ID dari pengguna yang login</p> <p>Menyimpan ID dari visite yang dipilih</p> <p>Melakukan koneksi ke web service konfirmasi visite dengan parameter ID pengguna dan ID visite</p> <p>Mendapatkan balasan berupa JSON bahwa status visite sudah diperbarui</p> <p>selesai</p>

5.8.9 Fungsi Status Operasi

Fungsi proses operasi dilakukan dengan memberikan *command* kepada basis data melalui *web service* untuk melakukan *update* kepada status operasi. Hal ini dicapai dengan mendapatkan nomor identitas pengguna dan nomor *ID* dari operasi yang dipilih dari proses pengambilan informasi detil operasi, lalu kemudian menjadikan nomor pengguna dan nomor operasi sebagai parameter untuk mengakses *web service*.

Pseudocode: Status Operasi

Mulai

Menyimpan ID dari pengguna yang login

Menyimpan ID dari operasi yang dipilih

Menyimpan pilihan status operasi (selesai, ditunda, tidak selesai)

Melakukan koneksi ke web service status operasi dengan parameter ID pengguna, ID operasi, dan status operasi

Mendapatkan balasan berupa JSON bahwa status operasi sudah diperbarui selesai



BAB 6 PENGUJIAN & ANALISIS

6.1 Pengujian

Pada bagian ini akan dijabarkan tahap-tahap dan hasil pengujian aplikasi. Pengujian yang dilakukan adalah pengujian *blackbox* untuk mendapatkan validitas fungsional aplikasi agar sesuai dengan perancangan dan memiliki keluaran yang sesuai dan pengujian *usability* untuk mendapatkan tingkat penerimaan aplikasi pada calon pengguna.

6.2 Pengujian Fungsional

Dalam bagian ini akan dijelaskan tiap-tiap kasus uji berdasarkan kebutuhan fungsional sistem pada tahap perancangan beserta hasil uji dari kasus tersebut. Terdapat 9 buah kasus uji dimulai dari Tabel 6.1 untuk kasus uji *login*, Tabel 6.2 untuk kasus uji daftar visite, Tabel 6.3 untuk kasus uji daftar visite, Tabel 6.4 untuk kasus uji daftar operasi, Tabel 6.5 untuk kasus uji detil pasien, Tabel 6.6 untuk kasus uji detil visite, Tabel 6.7 untuk kasus uji detil operasi, Tabel 6.8 untuk kasus uji konfirmasi visite, dan Tabel 6.9 untuk kasus uji status operasi.

6.2.1 Kasus Uji Login

Tabel 6.1 Kasus uji login

Nomor Kasus Uji	KF_01
Nama Kasus Uji	Login
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman menu jika username dan password yang dimasukkan sesuai dengan yang ada pada basis data.
Skenario untuk melakukan Login	
Prosedur Pengujian	Pengguna memasukkan username dan password Pengguna menekan tombol login
Hasil yang diharapkan	Pengguna masuk ke halaman menu
Hasil yang didapatkan	Pengguna masuk ke halaman menu
Status	Valid

6.2.2 Kasus Uji Melihat Daftar Pasien

Tabel 6.2 Kasus uji melihat daftar pasien

Nomor Kasus Uji	KF_02
Nama Kasus Uji	Melihat daftar pasien
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman daftar pasien yang sesuai, ketika menekan tombol pasien pada menu utama.
Skenario untuk melihat daftar pasien	
Prosedur Pengujian	Pengguna memilih tombol pasien pada menu utama.
Hasil yang diharapkan	Pengguna masuk ke halaman daftar pasien yang sesuai.
Hasil yang didapatkan	Pengguna masuk ke halaman daftar pasien yang sesuai.
Status	Valid

6.2.3 Kasus Uji Melihat Daftar Visite

Tabel 6.3 Kasus uji melihat daftar visite

Nomor Kasus Uji	KF_03
Nama Kasus Uji	Melihat daftar visite
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman daftar visite yang sesuai, ketika menekan tombol visite pada menu utama.
Skenario untuk melihat daftar visite	
Prosedur Pengujian	Pengguna memilih tombol visite pada menu utama.
Hasil yang diharapkan	Pengguna masuk ke halaman daftar visite yang sesuai.

Hasil yang didapatkan	Pengguna masuk ke halaman daftar visite yang sesuai.
Status	Valid

6.2.4 Kasus Uji Melihat Daftar Operasi

Tabel 6.4 Kasus uji melihat daftar operasi

Nomor Kasus Uji	KF_04
Nama Kasus Uji	Melihat daftar operasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman daftar operasi yang sesuai, ketika menekan tombol operasi pada menu utama.
Skenario untuk melihat daftar operasi	
Prosedur Pengujian	Pengguna memilih tombol operasi pada menu utama.
Hasil yang diharapkan	Pengguna masuk ke halaman daftar operasi yang sesuai.
Hasil yang didapatkan	Pengguna masuk ke halaman daftar operasi yang sesuai.
Status	Valid

6.2.5 Kasus Uji Melihat Detil Pasien

Tabel 6.5 Kasus uji melihat detil pasien

Nomor Kasus Uji	KF_05
Nama Kasus Uji	Melihat detil pasien
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman detil pasien yang sesuai dengan pasien terpilih.
Skenario untuk melihat detil pasien	
Prosedur Pengujian	Pengguna memilih salah satu pasien dari halaman daftar pasien.

Hasil yang diharapkan	Pengguna masuk ke halaman detil pasien yang sesuai.
Hasil yang didapatkan	Pengguna masuk ke halaman detil pasien yang sesuai.
Status	Valid

6.2.6 Kasus Uji Melihat Detil Visite

Tabel 6.6 Kasus uji melihat detil visite

Nomor Kasus Uji	KF_06
Nama Kasus Uji	Melihat detil visite
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman detil visite yang sesuai dengan visite terpilih.
Skenario untuk melihat detil visite	
Prosedur Pengujian	Pengguna memilih salah satu visite dari halaman daftar visite.
Hasil yang diharapkan	Pengguna masuk ke halaman detil visite yang sesuai.
Hasil yang didapatkan	Pengguna masuk ke halaman detil visite yang sesuai.
Status	Valid

6.2.7 Kasus Uji Melihat Detil Operasi

Tabel 6.7 Kasus uji melihat detil operasi

Nomor Kasus Uji	KF_07
Nama Kasus Uji	Melihat detil operasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa pengguna akan berpindah ke halaman detil operasi yang sesuai dengan operasi terpilih.
Skenario untuk melihat detil operasi	
Prosedur Pengujian	Pengguna memilih salah satu operasi dari halaman daftar operasi.

Hasil yang diharapkan	Pengguna masuk ke halaman detil operasi yang sesuai.
Hasil yang didapatkan	Pengguna masuk ke halaman detil operasi yang sesuai.
Status	Valid

6.2.8 Kasus Uji Konfirmasi Visite

Tabel 6.8 Kasus uji konfirmasi visite

Nomor Kasus Uji	KF_08
Nama Kasus Uji	Konfirmasi visite
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa status visite dokter untuk pasien tertentu sudah berubah pada basis data ketika pengguna memilih tombol konfirmasi pada halaman detil visite.
Skenario untuk konfirmasi visite	
Prosedur Pengujian	Pengguna memilih tombol konfirmasi pada halaman detil visite.
Hasil yang diharapkan	Status visite pasien yang sesuai berubah pada basis data.
Hasil yang didapatkan	Status visite pasien yang sesuai berubah pada basis data.
Status	Valid

6.2.9 Kasus Uji Status Operasi

Tabel 6.9 Kasus uji status operasi

Nomor Kasus Uji	KF_09
Nama Kasus Uji	Status Operasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian bertujuan untuk memastikan bahwa status operasi akan berubah pada basis data sesuai dengan apa yang dipilih pengguna pada halaman detil operasi.
Skenario untuk status operasi	

Prosedur Pengujian	Pengguna memilih tombol status operasi pada halaman detil operasi Pengguna memilih salah satu pilihan status yang tersedia
Hasil yang diharapkan	Status operasi pasien berubah pada basis data sesuai dengan pilihan pengguna.
Hasil yang didapatkan	Status operasi pasien berubah pada basis data sesuai dengan pilihan pengguna.
Status	Valid

6.3 Hasil Pengujian Fungsional

Berdasarkan pengujian untuk tiap-tiap kebutuhan fungsional aplikasi, didapatkan bahwa seluruh fungsi berjalan sesuai dengan perancangan dan memiliki keluaran yang diharapkan. Penggunaan aplikasi dengan prosedur yang sudah ditetapkan, tidak memiliki kesalahan dalam memberikan keluaran.

6.4 Pengujian Usabilitas

Untuk pengujian usabilitas, digunakan metode SUS (*System Usability Scale*) yaitu metode untuk menguji usabilitas dengan cepat dan dapat diandalkan (Brooke, 1996). Dalam metode SUS, responden diberikan 10 pertanyaan yang memberikan pandangan umum tentang *usability* dari sebuah sistem. Responden dapat menjawab pertanyaan dalam 5 tingkatan, mulai dari sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju.

Tipe pertanyaan pada metode SUS adalah berselang-seling antara pertanyaan positif dan negatif, yang tiap jenis pertanyaan memiliki kriteria penilaian tersendiri. Pertanyaan positif akan dihitung dengan cara mengurangi bobot jawaban dengan 1, yaitu $(x-1)$ dimana x adalah jawaban untuk pertanyaan positif. Sedangkan pertanyaan negatif akan dihitung dengan cara mengurangi 5 dengan bobot jawaban, yaitu $(5-y)$ dimana y adalah bobot jawaban pada pertanyaan negatif. Total bobot untuk semua pertanyaan kemudian dikali dengan 2,5 untuk mendapatkan nilai akhir SUS dalam jangka 0-100.

Dalam pengujian aplikasi ini, pertanyaan yang diberikan seperti yang ada pada Tabel 6.10. Pengujian dilakukan kepada 5 orang responden yang merupakan tenaga medik yang pernah mengikuti proses visite atau operasi pada rumah sakit. 5 orang responden mencoba aplikasi terlebih dahulu, untuk kemudian diberikan kuesioner *usability* aplikasi.

Tabel 6.10 Pertanyaan pengujian *usability* SUS

No	Pernyataan	Sangat tidak setuju					Sangat setuju				
1.	Saya merasa bahwa saya ingin lebih sering menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Saya merasa bahwa aplikasi ini terlalu rumit.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Saya pikir aplikasi ini mudah digunakan.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Saya berpikir bahwa saya akan membutuhkan bantuan dari seorang ahli untuk dapat menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Saya menganggap berbagai fitur di aplikasi ini terintegrasi dengan baik.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Saya pikir terlalu banyak ketidakkonsistenan di aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	Saya pikir bahwa kebanyakan orang akan dapat belajar menggunakan aplikasi ini dengan cepat.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Saya merasa aplikasi ini merepotkan untuk digunakan.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Saya merasa sangat percaya diri saat menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Saya perlu belajar banyak hal terlebih dahulu sebelum menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

6.5 Hasil Pengujian Usabilitas

Setelah memberikan kuesioner pertanyaan kepada 5 orang responden, didapatkan hasil pembobotan jawaban seperti yang tertulis pada Tabel 6.11, lalu dilakukan perhitungan untuk setiap bobot pertanyaan yang diberikan. Dengan menggunakan persamaan perhitungan yang tercantum pada bab 2 tentang SUS, didapatkan hasil seperti yang tertera pada Tabel 6.12.

Tabel 6.11 Hasil Pembobotan Jawaban Tiap Responden

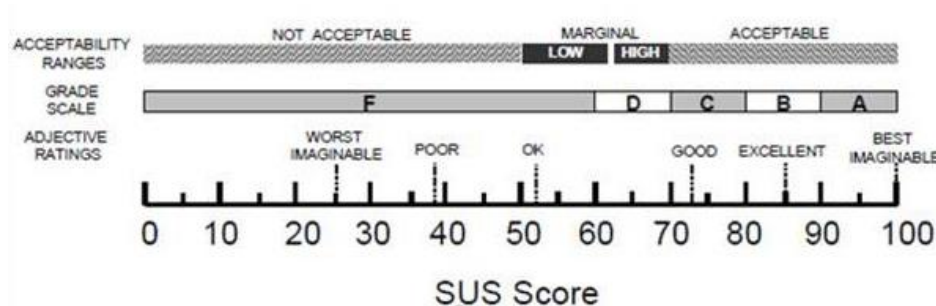
Nomor pertanyaan	Skor Responden 1	Skor Responden 2	Skor Responden 3	Skor Responden 4	Skor Responden 5
1	3	3	3	3	2
2	3	3	3	3	3
3	4	4	3	4	3
4	3	4	4	4	3
5	3	3	2	3	4
6	4	4	3	3	2
7	3	4	3	3	4
8	3	3	4	4	3
9	2	3	2	2	2
10	3	4	4	3	3
Total Skor	31	35	32	32	29

Tabel 6.12 Hasil Perhitungan Skor SUS

Nomor Responden	Total Skor SUS
Responden 1	77,5
Responden 2	87,5
Responden 3	80
Responden 4	80
Responden 5	72,5
Rata-rata	79,5

6.6 Analisis Hasil Pengujian Usabilitas

Berdasarkan hasil rata-rata skor SUS dapat dilakukan penilaian dengan tingkat *Not Acceptable* untuk skor 0-50.9, *Marginal* untuk rentang skor 51-70.9, dan *Acceptable* 71-100(Ardyansiah, 2016) seperti yang tertera pada Gambar 6.1.



Gambar 6.1 Penilaian Skor SUS

Dari hasil perhitungan skor SUS untuk setiap responden didapatkan rata-rata skor sejumlah **79,5** dengan tingkat penilaian *Acceptable*, yang berarti aplikasi sudah dapat diterima oleh calon pengguna untuk diimplementasikan.

6.7 Pengujian Waktu

Untuk memastikan bahwa implementasi aplikasi memenuhi tujuan penelitian, yaitu mempercepat waktu yang diperlukan tenaga medis untuk mendapatkan informasi pasien visite dan operasi, maka dilakukan pengujian waktu untuk membandingkan lama aktivitas sebelum implementasi dan setelah implementasi aplikasi untuk proses yang sama.

Pada pengujian ini dilakukan perbandingan untuk waktu mendapatkan detail visite seorang pasien. Untuk waktu sebelum implementasi didapatkan dari proses wawancara dengan *stakeholder* Rumah Sakit XYZ, dan waktu setelah implementasi didapatkan dengan menghitung rata-rata waktu yang dibutuhkan oleh 5 orang responden uji untuk mendapatkan informasi detail visite. Hasil pengujian seperti yang tertera pada Tabel 6.13.

Tabel 6.13 Hasil Pengujian Waktu Responden

Nomor Responden	Waktu(detik)
1	16.18
2	20.25
3	15.06
4	18.35
5	16.89
Rata-rata	17.346

Waktu yang dibutuhkan oleh tenaga medis untuk mendapatkan informasi visite pasien berdasarkan wawancara adalah 3 menit(180 detik), maka implementasi sistem dengan waktu rata-rata 17 detik mempercepat proses mendapatkan informasi sejumlah 163 detik(900% lebih cepat). Hasil ini berarti kebutuhan mempercepat proses sudah teruji dan terbukti.



BAB 7 KESIMPULAN

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, didapatkan 9 buah kebutuhan fungsional aplikasi yang didapatkan dari wawancara dengan stakeholder Rumah Sakit XYZ. Kebutuhan fungsional sistem yang dibangun memiliki 3 fitur utama yaitu fitur pasien, visite, dan operasi.

Hasil perancangan yaitu berupa sequence diagram untuk gambaran proses sistem berdasarkan waktu, class diagram untuk relasi antar objek pada sistem, struktur basis data untuk gambaran hubungan entitas pada basis data yang digunakan, dan perancangan antar muka.

Hasil implementasi sistem berdasarkan kepada proses perancangan, implementasi kode tiap proses fungsi sesuai dengan perancangannya. Hasil implementasi antarmuka sesuai dengan perancangan mockup.

Proses pengujian aplikasi menggunakan metode blackbox dengan hasil valid untuk 9 buah kasus uji yang didasarkan pada setiap kebutuhan fungsional. Pengujian usabilitas menggunakan metode SUS(System Usability Scale) mendapatkan skor rata-rata 79.5 untuk 5 orang responden, yang berarti usabilitas sudah berada dalam kriteria acceptable atau sudah dapat diterima calon pengguna. Pengujian waktu dengan membandingkan waktu sebelum implementasi aplikasi dengan rata-rata waktu setelah implementasi aplikasi untuk mendapatkan informasi yang sama menghasilkan percepatan proses dari 180 detik menjadi 17 detik, yang berarti implementasi aplikasi ini mempercepat proses alur informasi pasien untuk tenaga medis di Rumah Sakit XYZ.

7.2 Saran

Untuk penelitian berikutnya diharapkan dapat menambahkan fungsi pencarian untuk bagian pasien, visite, maupun operasi. Fitur notifikasi juga dapat diimplementasikan untuk memudahkan tenaga medis dalam mengingat waktu-waktu penting untuk visite dan operasi. Juga agar penelitian ini dapat dijadikan landasan awal untuk pengembangan sistem informasi *mobile* untuk tenaga medis terutama dokter, karena proses bisnis institusi kesehatan terutama yang melibatkan dokter merupakan salah satu proses yang terpenting dan dapat berpengaruh terhadap kesehatan dan keselamatan pasien.

DAFTAR PUSTAKA

- Ardiansyah, 2016. Pengujian Usability User Interface dan User Experience Aplikasi E-Reader Skripsi Berbasis Hypertext. S1. Universitas Ahmad Dahlan.
- Brooke, John., 1995. *SUS: A quick and dirty usability scale*. San Fransisco: Morgan Hauffman
- e27, 2015. Infographic: Android Dominates SEA Asia's Smartphones.(<https://e27.co/infographic-android-dominates-southeast-asias-smartphones-20151026/>)
- Kadir, A., 2003. *Pengenalan Sistem Informasi*. Yogyakarta:Andi Offset.
- Khalifa, M., 2015. *Hospital Information Systems (HIS) Acceptance and Satisfaction: A Case Study of a Tertiary Care Hospital*. MD. The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare.
- Nielsen, J., 1993. *Usability Engineering*. San Fransisco: Morgan Hauffman.
- Rogers, R., Zigurd, M., Hansson, J., Laird, D. & Blake, G., 2011. *Programming Android: Java Programming for the New Generation of Mobile Devices*. Canada: O'Reilly Media.
- Salis, F, 2016. *Sistem Informasi Visite Dokter Rawat Inap Pada Rumah Sakit Umum Islam Harapan Anda*. S1. Universitas Dian Nuswantoro.
- Sommerville, I., 2011. *Software Engineering*, 9th ed. Boston, Massachusetts: Pearson Education.